

Ein Neigungsmesser - Array an der KTB

-

Datenbearbeitung und -auswertung der Messwerte
der Messungen mit fünf ASKANIA-
Bohrlochneigungsmessern

Diplomarbeit

zur Erlangung des akademischen Grades Diplom-Geophysiker



Friedrich-Schiller-Universität Jena

Chemisch - Geowissenschaftliche Fakultät
Institut für Geowissenschaften, Lehrstuhl für Angewandte Geophysik

eingereicht von André Gebauer
geb. am 09.11.1978 in Sömmerda

Jena, im Februar 2006

Betreuer:

Prof. Gerhard Jentsch

Dr. Thomas Jahr

A. Gebauer: Ein Neigungsmesser-Array an der KTB.

Errata

Aufgrund eines Bearbeitungsfehlers sind einige der gezeigten Resultate um den Faktor 10 zu klein.

Daher sind die gezeigten Ergebnisse auf den Seiten 120 (Abb. 8.16), 121 (Abb. 8.18), sowie die dazugehörige Beschreibung im Text, Kapitel 8.5 „Hodogramme“ mit 10 zu multiplizieren. Dies betrifft ebenso die im Kapitel 9 „Interpretation und Diskussion“, Unterpunkt „Driften und Hodogramme“ gezeigten Abbildungen und Aussagen.

Jena, 01.07.2006

André Gebauer

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation, Aufgabenstellung	2
1.2	Beteiligte Parteien – Partner, Kooperationen.....	3
1.3	Geplanter Verlauf der Injektion an der Vorbohrung.....	3
1.4	Theoretische Vorbetrachtungen	5
1.5	Voruntersuchungen	8
2	Aufbau und Struktur des Neigungsmesser-Arrays	10
2.1	Auswahl der Standorte	10
2.2	Ausführung der Bohrungen.....	12
2.3	Das Funknetzwerk zur Datenübertragung (WLAN).....	16
3	Geologie	17
3.1	Der weite geologische Rahmen.....	17
3.2	Der engere geologische Rahmen.....	18
3.3	Das geologische Umfeld der KTB mit den Neigungsmesserstationen	21
4	Grundlagen	23
4.1	Neigungsmesser	23
4.2	Signalgehalt von Neigungsmessungen.....	24
4.3	Gezeitenpotential	26
4.4	Poroelelastische Rheologie und punktförmige Fluidinjektion.....	31
5	Messgeräte, Durchführung der Messungen und Sicherung der Daten	38
5.1	ASKANIA-Bohrlochneigungsmesser	38
5.1.1	Aufbau	39
5.1.2	Ankopplung	43
5.1.3	Kalibriervorrichtung	44
5.1.4	Azimetbestimmung.....	46
5.1.5	Auflösung, Linearität, Langzeitstabilität der Registrierungen.....	48
5.1.6	Übertragungseigenschaften.....	48
5.1.7	Registrierung der Daten	49
5.2	Diver – Grundwasser-Messsensoren.....	51
5.3	Meteorologische Station	55
6	Datenauswertung	57

6.1	Allgemeiner Überblick über die Auswertung von Tiltmeter-Daten	57
6.2	Auswertung von Tiltmeter-Daten	58
6.2.1	Datenaufbreitung	59
6.2.1.1	Programm – <i>DatAdd.exe</i>	60
6.2.1.2	Programm – <i>PreAnalyse.exe</i>	64
6.2.1.3	Programm – <i>EternaAdd.exe</i>	74
6.2.2	Filtern der Datenreihen – <i>Filter.exe</i>	76
6.2.3	Kalibrierfaktoren – Kalibrieren der Datenreihen	79
6.2.3.1	Bestimmung der Kalibrierfaktoren	79
6.2.3.2	Kalibrieren der Daten – <i>Kalibrierung.exe</i>	81
6.2.4	Transformieren der Datenreihen – <i>Transformation.exe</i>	81
6.2.4.1	Ändern des Datenformates – <i>ChFormat.exe</i>	84
6.2.5	Gezeitenanalyse – <i>Analyze.exe</i>	85
6.3	Darstellen des Datenauswertungssystems	88
6.4	Aufbereitung der Wetter- und Grundwasserdaten	94
7	Verlauf der Injektion	97
8	Datengrundlage und -analyse	99
8.1	Datengrundlage	99
8.2	Analyse der Gezeitenparameter	105
8.3	Driften der Zeitreihen	109
8.4	Luftdruck	115
8.5	Hodogramme	118
8.6	Das Sumatra-Andamanen-Beben vom 26.12.2004	123
9	Interpretation und Diskussion	125
10	Zusammenfassung	137
	Danksagung	138
	Literatur	139
	Anhang	143
I	Liste der Symbole	143
II	Programmcode	145
II.I	Programmcode - <i>DatAdd.exe</i>	145
II.II	Script der Batch-Datei <i>DatAdd.bat</i>	150
II.III	Programmcode - <i>PreAnalyse.exe</i>	154
II.IV	Programmcode - <i>EternaAdd.exe</i>	160
II.V	Programmcode - <i>Filter.exe</i>	162
II.VI	Programmcode - <i>Kalibrierung.exe</i>	171
II.VII	Programmcode - <i>Transform.exe</i>	171
III	Steuerfile für <i>Analyse.exe</i>	172
IV	Aufstellung der Verwendeten Batchdateien und Programme	174
V	Aufstellung der vorhandenen Neigungsmesser- und Grundwasserdaten	176

1 Einführung

In der unmittelbaren Nähe von Windischeschenbach in der Oberpfalz entstand im Rahmen des International Continental Deep Drilling Project (ICDP) die Kontinentale Tiefbohrung der Bundesrepublik (KTB) (Abb. 1.1) im Zeitraum von 1987 bis 1994. Zur KTB gehören die Vorbohrung (VB) mit einer Teufe von 4001 m, welche bis 3750 m unverohrt ist und die Hauptbohrung mit einer Teufe von 9101 m.



Abb. 1.1: Lage der KTB in Deutschland.

Nach zahlreichen Versuchen nach Beendigung der Bohrphase fand 2004/05 ein großskaliger Injektionsversuch statt. Dabei wurde Wasser in die Vorbohrung injiziert. Die Bandbreite der Untersuchungen in den Geowissenschaften während des Versuches sind zahlreich (Kapitel 1.2).

1.1 Motivation, Aufgabenstellung

In den letzten Jahrzehnten gab es einige Versuche, bei denen mit Hilfe von Neigungsmessern die Deformation der Erdkruste durch Injektion von Fluiden (Kümpel 1988; Kümpel, 1989; Kümpel 1996; Weise et al., 1999; Lehmann, 2001; Fabian, 2004; u.a.) registriert wurde. Diese Versuche hatten Dimensionen von wenigen 10er m. An der KTB sollte dieses Experiment wiederholt werden mit einer Dimension im Kilometer-Maßstab. Die hoch auflösenden Neigungsmessungen sollen im besten Fall, wie in Abb. 1.2 zu sehen, die Deformation messen.

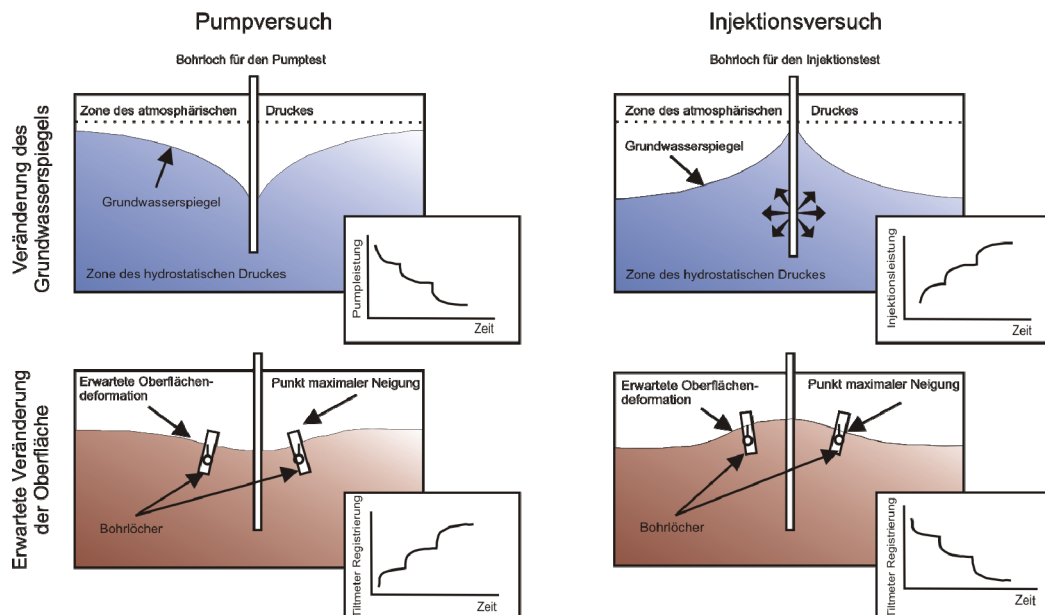


Abb. 1.2: Darstellung der Oberflächenverformung und des erwarteten Neigungssignales bei Pump- und Injektionsversuchen.

Nach der Genehmigung des Projektes durch die DFG konnten die 5 Neigungsmesser installiert werden. Danach stellte sich sehr schnell heraus, dass die enorme Datenflut sich nicht mit den bisherigen Mitteln bewältigen ließ. Somit musste ein System entwickelt werden, um die Datenbearbeitung zu automatisieren. Die Darstellung des Aufbereitungs- und Auswertungssystems sowie der Ergebnisse und ersten Interpretationen soll Gegenstand dieser Arbeit sein.

1.2 Beteiligte Parteien – Partner, Kooperationen

Die Verbindungen zu anderen Wissenschaftlern gehen um die ganze Welt. Zum einen gibt es die Partner im Rahmen des Projektes und zum anderen internationale Kontakte.

Im Rahmen des Projektes bestehen neben den Antragstellern aus der Geologie, Universität Göttingen, enge Verknüpfungen mit dem Geoforschungszentrum Potsdam (Installation von Geophonen auf unseren Neigungsmessern zur Registrierung seismischer Signale, gemeinsame Datenübertragung, Gruppe Asch und Mitantragsteller) und dem GGA-Institut Hannover (gemeinsame Bearbeitung und Auswertung der Grundwasser-Mess-Stellen an den Bohrungen der Neigungsmesser, Kessels et al., Kolditz et al.). Weitere Verbindungen ergeben sich zu den Projekten in der Geoelektrik (Jacobs) und Hydrogeochemie.

Internationale Kontakte bestehen hinsichtlich der Interaktion von Fluiden und geophysikalischen Prozessen. In erster Linie sind hier zu nennen Prof. H. Ishii, TRIES, Mizunami-city, Japan, der ebenfalls schon Injektionsversuche in Bohrlöchern durchgeführt hat und Herr Prof. A. Špičák, Institut für Geophysik der Tschechischen Akademie der Wissenschaften, Prag, der unser Partner bei den Arbeiten im Schwarmbeben-Gebiet Vogtland/NW-Böhmen ist. Weiterhin wird verwiesen auf die Beteiligung im Global Geodynamics Project (GGP), in dem u.a. auch die Auswirkungen von Bodenwässern auf den Schwerevektor in allen Skalen untersucht werden.

1.3 Geplanter Verlauf der Injektion an der Vorbohrung

Der folgende Injektionsplan (Tab. 1.1, Abb. 1.3) wurde bei dem Treffen aller Antragsteller nach Ihren jeweiligen Bedürfnissen am 15.05.2003 aufgestellt (Sitzungsprotokoll). Dabei wurden folgende Schwerpunkte gesetzt:

- Beobachtung des Kaiser-Effekts¹
- Erzielung eines möglichst hohen Druckgradienten
- Injektion eines möglichst hohen Gesamtvolumens

¹ Kaiser-Effekt: Bei Belastungsversuchen an festen Körpern treten Schallemissionen auf. Der Effekt, dass der Körper erst dann Schall emittiert, wenn er über das bisherige maximale Spannungsniveau hinaus belastet wird, wird Kaiser-Effekt bezeichnet. Dieser Effekt wurde von J. Kaiser (1950) beobachtet.

- Injektionsphasen außerhalb der Gezeitenperioden
- ...

Tab. 1.1: Geplanter Verlauf der Injektion (Sitzungsprotokoll).

Dauer	Druck in MPa
3 Tage	4
3 Tage	8
3 Tage	12
3 Tage	16
10 Tage	0 (shut in)
3 Tage	4
3 Tage	8
3 Tage	12
3 Tage	16
10 Tage	0 (shut in)
2 Tage - 5 Zyklen á:	
5 Stunden	16
5 Stunden	0 (shut in)
4 Monate	16
2 Monate	8
2 Monate	16
1 Monat	8
1 Monat	16
15 Tage	0 (shut in)
3 Tage	4
3 Tage	8
3 Tage	12
3 Tage	16

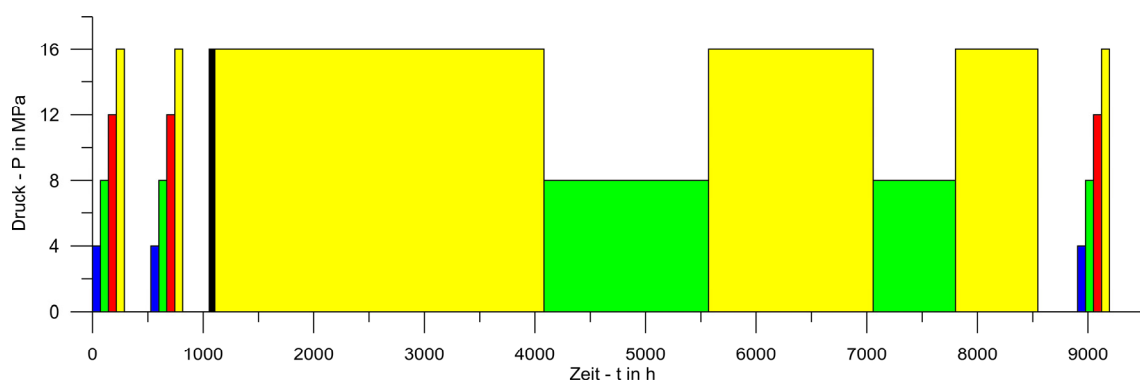


Abb. 1.3: Ursprünglich Geplanter Verlauf der Injektion.

1.4 Theoretische Vorbetrachtungen

Anhand des geplanten Injektionsverlaufes wurden von Jahr et al. (2005, 2006) während der Vorbereitungsphase Modellierungen vorgenommen. Diese waren zum einen notwendig, um eine optimale Entfernung der Neigungsmesserstationen zu ermitteln und zum anderen, um die Größenordnung des Mess-Signals abzuschätzen.

Die nachfolgenden Ergebnisse wurden mit Hilfe des Modellierungsprogramms *POEL* erstellt. Dabei wird angenommen, dass eine Pumpleistung von 200 Litern pro Minute und ein Druck von 16 MPa erreicht würden. Das von Wang (2002) entwickelte Programm legt eine poroelastische Rheologie zugrunde und setzt einen geschichteten Untergrund voraus (s. Abb. 1.4).

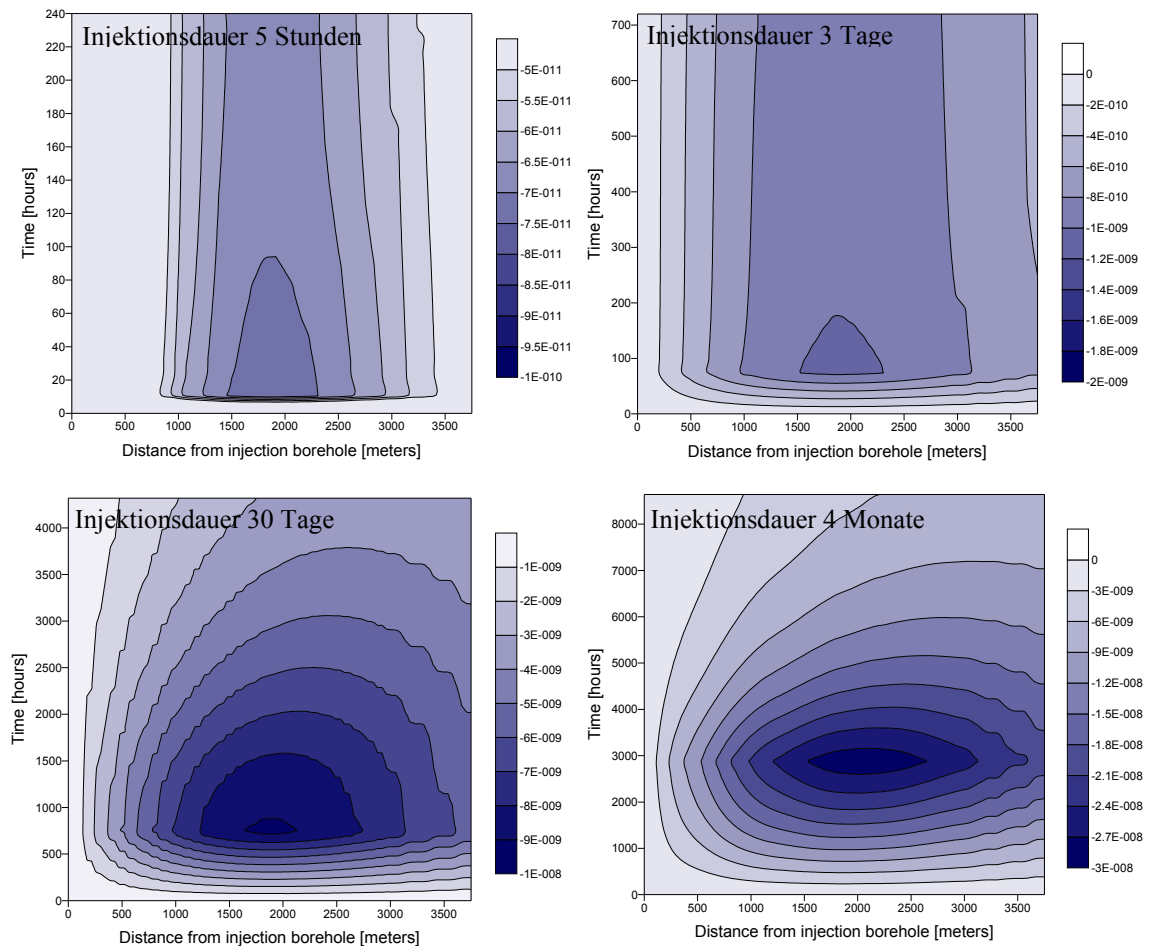


Abb. 1.4: Resultate der Modellierung mit *POEL*. Die Neigungseffekte (in Rad) sind hier für verschiedene Injektionsintervalle zu sehen (nach Jahr et al., 2005).

In Abb. 1.4 sind jeweils die erwarteten radialen Neigungen in Radian für ein Injektionsintervall dargestellt, wobei der zeitliche Ablauf über der Entfernung zum Injektionspunkt aufgetragen ist.

Für ein Injektionsintervall von 5 h ergeben sich maximale radiale Neigungen von 0.1 nrad. Diese zusätzliche Driftkomponente, die nach ca. 20 h auftritt, liegt somit unterhalb der Nachweisgrenze der Neigungsmesser. Mit zunehmender Injektionszeit verschiebt sich der Bereich der maximalen Neigungen. Bei einer Injektionszeit von 4 Monaten wird nach knapp 3000 h eine induzierte Neigung von ca. 30 nrad modelliert. Dieses zusätzliche Driftsignal sollte dann an den Stationen der Neigungsmesser beobachtet werden können. Die Amplitude liegt bei knapp der Hälfte der maximalen Gezeitenamplitude und sollte somit signifikant nachweisbar sein.

Die durch Langzeit-Injektionen bewirkten Neigungen, die als zusätzliche Driften in den Registrierungen vorhanden sein werden, sollten nachweisbar sein. Der induzierte Driftanteil ist unter folgenden Bedingungen separierbar:

- Es muss ein möglichst hoher Injektionsdruck von mindestens 16 MPa aufgebaut werden.
- Die lokal verursachten Driftanteile, z.B. durch Grundwasser-Variationen, an den einzelnen Stationen müssen möglichst gut kontrolliert und ggf. korrigiert werden.
- Die weltweit erstmalig für diese Neigungsmesser eingesetzte Array-Technik ermöglicht das zeitgleiche Beobachten des induzierten Neigungseffekts an allen fünf Neigungsmesserstationen, d.h. der parallele, kontinuierliche Betrieb der Stationen muss gewährleistet sein.

Ein weiterer Modellierungsansatz wurde unter *ABAQUS* mit einem Finiteelement Model (FEM) (Abb. 1.5) vorgenommen (Fischer, 2002)¹. Für das FEM wurde die poroelastische Rheologie (siehe Kapitel 4) angewendet. Die KTB befindet sich in der Mitte des FEM, welches eine modellierte Größe von 10x10x10 km hat. Der Injektionspunkt befindet sich dabei in einer Tiefe von 4 km. Die Annahmen dafür sind, dass der Untergrund homogen ist. Als Strukturelement wurde die an der KTB befindliche Hauptstörungszone implementiert. Die Injektion wird über 1 Jahr mit einem Druck von 16 MPa angesetzt. In Abb. 1.6 sind die Ergebnisse der Verschiebungen dargestellt. Die

¹ Persönliche Mitteilung

größten Verschiebungen herrschen am Injektionspunkt und an der Erdoberfläche in einem Abstand von 2.4 km. Aus dieser Erkenntnis und aus der Modellierung mit *PEOL* konnte ein optimaler Abstand der Beobachtungsorte von der KTB von 2.4 km ermittelt werden. Weitere Informationen zu den Modellierungen sind bei Jahr et al. (2005, 2006) zu finden.

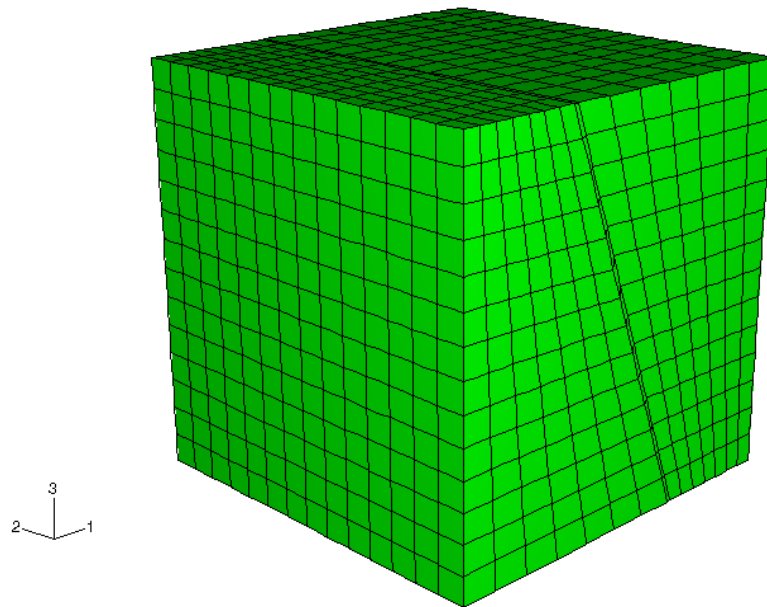


Abb. 1.5: Finiteelementmodel (*ABAQUS*) (Fischer, 2002)¹.

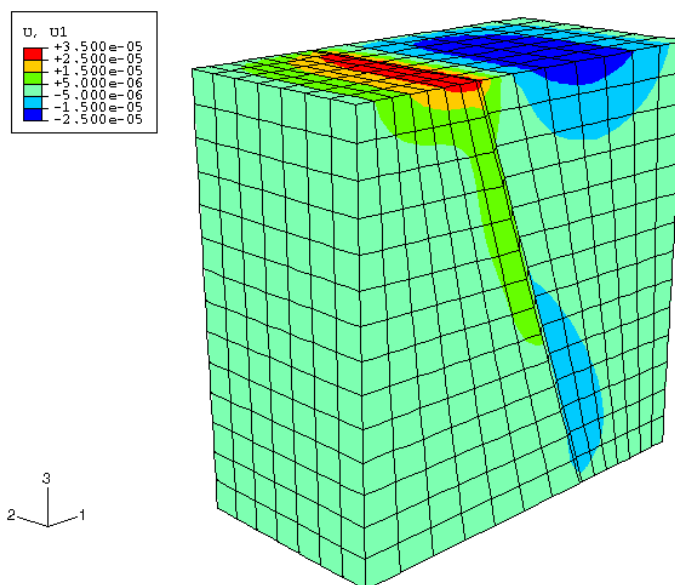


Abb. 1.6: Modellierung der Verschiebungen mit *ABAQUS* (Injektionsdauer 1 Jahr mit einem Druck von 16 MPa) (Fischer, 2002)¹.

¹ Persönliche Mitteilung

1.5 Voruntersuchungen

Nachdem mit Hilfe der Modellierungen der optimale Abstand der Messorte bestimmt worden war, ging es darum, entsprechende Standorte zu finden. Jede Station sollte eine Anbindung an das Stromnetz besitzen und einen Ort haben, an dem die um den Neigungsmesser zu betreibenden notwendigen Geräte (Kapitel 5.1.7) trocken und wenn möglich temperaturkonstant untergestellt werden können, sowie eine Sichtverbindung zum Turm der KTB, damit die Funkstrecke für die Datenübertragung eingerichtet werden konnte. Aber das wichtigste Kriterium für die Wahl des Standortes war der Untergrund (Kapitel 5.1.4). Die besten Messergebnisse erlangt man bei einer optimalen spannungsfreien Ankopplung des Messgerätes an den Untergrund und einem möglichst großen Abstand von der Oberfläche des Grundwasserhorizontes. Die beste Ankopplung der Geräte wird im Festgestein erreicht. Um diesen beiden Bedingungen gerecht werden zu können, wurden geoelektrische Voruntersuchungen unternommen. In Abb. 1.7 ist ein Bild dieser Untersuchungen zu sehen. Ein Ergebnis ist in Abb. 1.8 dargestellt. Dazu wurde das Verfahren der Tiefensondierung nach Schlumberger angewendet. Auf der linken Seite der Abbildung sind die gemessenen Werte und rechts die Ergebnisse mit verschiedenen möglichen Kurven gezeigt. Die rote Kurve markiert das am besten angepasste Modell. Die Kurve besagt, dass in einer Tiefe von 18 bis 20 m die Leitfähigkeit um zwei Größenordnungen sinkt. Dies ist ein Hinweis darauf, dass man in dieser Tiefe damit rechnen kann, auf das gewünschte Festgestein zu stoßen. Abb. 2.3. kann man entnehmen, dass man beim Bohren bei etwa 22 m Tiefe auf das Festgestein gestoßen ist.



Abb. 1.7: Voruntersuchungen mit Geoelektrik (Schlumberger Anordnung).

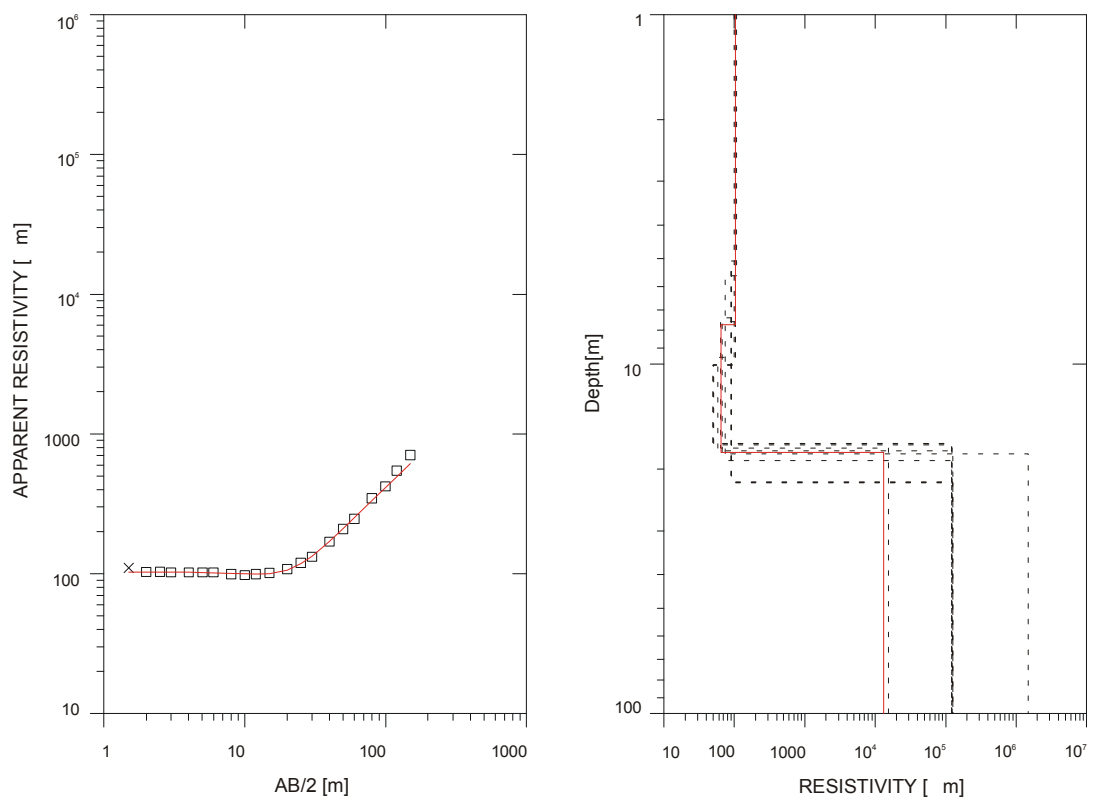


Abb. 1.8: Resultate der geoelektrischen Untersuchungen für die Station Berg (mit *Resix-P* ausgewertet).

2 Aufbau und Struktur des Neigungsmesser-Arrays

Nachdem die Standorte (Kapitel 2.1) der Neigungsmesser festgelegt waren, wurde von der Firma Pruy gebohrt. Die Bohrungen und Messergebnisse werden unter Kapitel 2.2 vorgestellt. Nach Abteufung der Bohrung und der Verrohrung wurden die Messinstrumente installiert. Auf dem Kopf der Neigungsmesser (Kapitel 5.1) wurden zur Registrierung seismischer Signale (Asch et al., 2003) vom GFZ bereit gestellte Geophone installiert. Um die Daten online übertragen zu können, wurde seitens des GFZ ein Funknetzwerk installiert (WLAN), das in Kapitel 2.3 vorgestellt wird.

2.1 Auswahl der Standorte

Auf Grund der Ergebnisse der Voruntersuchungen wurden die fünf Stationen Berg, Eiglasdorf Kapelle, Stockau, Mittelberg und Püllersreuth gewählt. In Abb. 2.1. ist die Lage der Stationen und die optimale Entfernung von der KTB aus Kapitel 1.3 eingezeichnet (Kreis). Zu sehen ist, dass die Stationen Eiglasdorf und Püllersreuth die ideale Entfernung haben. Stockau befindet sich außerhalb des Kreises mit dem Radius von 2.4 km, Mittelberg und Berg befinden sich innerhalb.

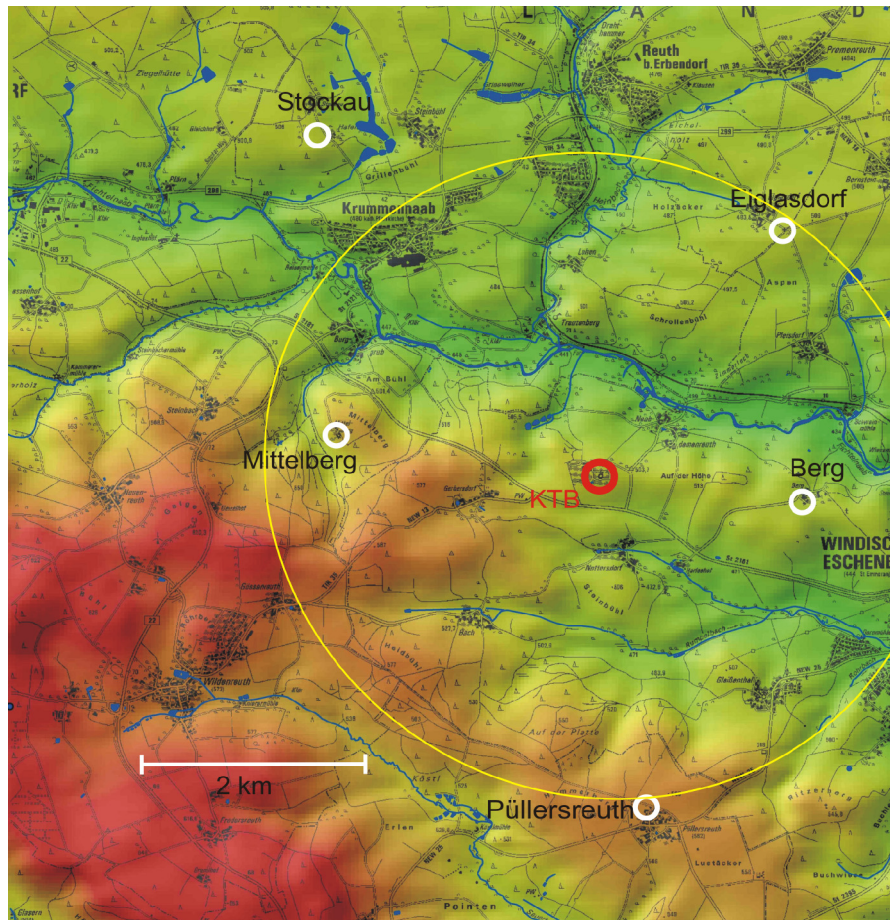


Abb. 2.1: Karte der Neigungsmesser-Stationen; der Kreis bezeichnet die ermittelte optimale Entfernung.

Die Vermessung der Stationen und Grundwassermess-Stellen wurde in drei Kampagnen, teilweise unter erschwerten Bedingungen (Abb. 2.2), durchgeführt. Das



Abb. 2.2: Vermessung Februar 2004 in Berg.

Messgerät befindet sich direkt über einem Trigonometrischen Punkt. Wie an dieser Station wurden alle Messnetze in Trigonometrischen Punkten, welche vom Bayerisches Landesvermessungsamt zur Verfügung gestellt wurden, eingehängt. Trotz der widrigen Umstände (Abb. 2.2) wurden bei dieser Kampagne sehr gute Ergebnisse erzielt. Diese sind in Tabelle 2.1 zusammen gestellt. Neben den Gauss-Krüger-Koordinaten der Stationen und der KTB sind in der Tabelle die Höhen über NN und die Entfernungen von der KTB zu entnehmen.

Tab. 2.1: Geographische Parameter.

Station	Geographische Koordinaten		Höhe über NN in m	Entfernung von der KTB in m
	Rechtswert	Hochwert		
KTB	4508772.1	5519865.1	601.600 (Turm- spitze)	
Berg	4510264.0	5519658.7	497.9	1590.0
Eiglasdorf	4510139.7	5521757.1	506.0	2390.0
Stockau	4506757.1	5522450.9	502.2	3230.0
Mittelberg	4506840.9	5520180.9	532.5	1870.0
Püllersreuth	4509099.3	5517301.7	547.0	2600.0

2.2 Ausführung der Bohrungen

Die Bohrlöcher für die Neigungsmesser und Grundwassermessstellen wurden von der Firma Pruy aus Schönheide mit einem kleinen Rigg (Abb.2.3) gebohrt.



Abb. 2.3: Das Bohrgerät der Firma Pruy an der zukünftigen Station Berg.

Aus den Aufzeichnungen von Letz (2004)¹ wurden die folgenden Bohrprofile (Abb. 2.3, 2.4, 2.5, 2.6, 2.7) erstellt. Die in den Abbildungen enthaltenen Ergebnisse der Vermessungen der Senkrechten der Bohrungen wurden von der Firma C & E Engineering & Consulting aus Chemnitz ermittelt und die Tiefen für den Ausbau der Bohrungen festgelegt. Diese Tiefen sind für die jeweiligen Stationen in den Bohrprofilen eingezeichnet.

Die Unterkante der Verrohrung, also die Unterkante des Einbaus des Neigungsmessers, ist in Tabelle 2.2 angegeben. Der Vollständigkeit halber sind in Tabelle 2.3 die Parameter der Grundwassermesspunkte angefügt (Kapitel 2.1). An den Stationen Berg, Stockau und Püllersreuth sind vom GGA-Institut jeweils zwei zusätzliche Grundwassermessstellen installiert worden, welche ebenso in Tabelle 2.3 angegeben sind.

¹ Persönlich Aufzeichnungen

2 Aufbau und Struktur des Neigungsmesser-Arrays

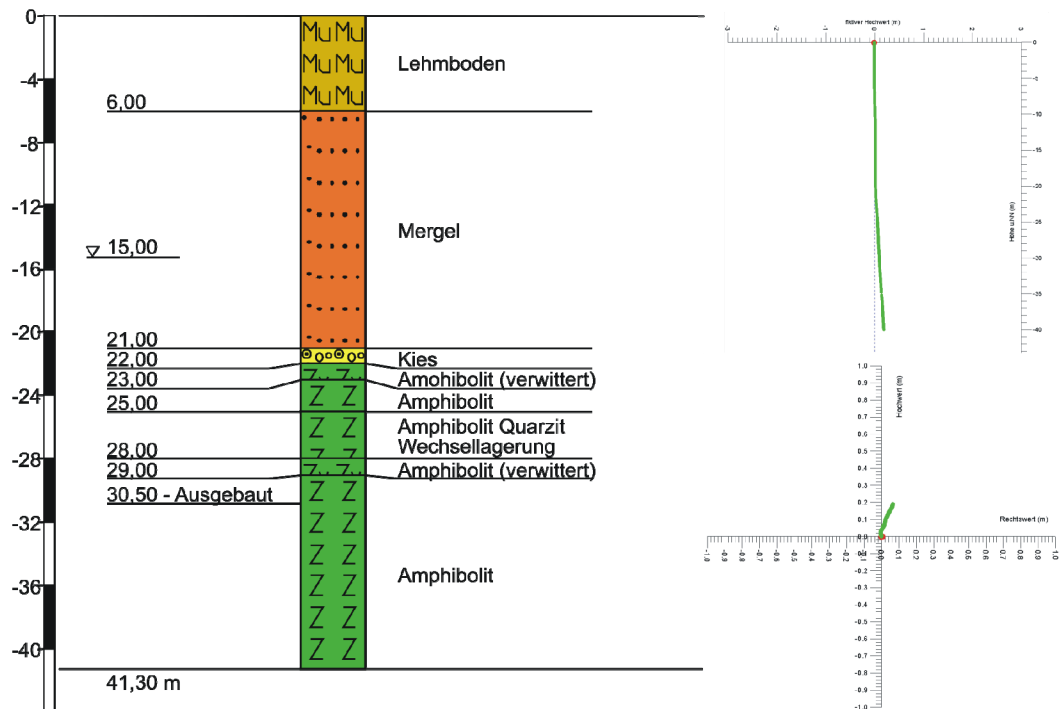


Abb. 2.4: Bohrprofil (links) und Ergebnisse der Abweichung aus der Vertikalen (vertikale Projektion – oben rechts; horizontale Projektion – unten rechts; C & E Engineering & Consulting) für die Station Berg.

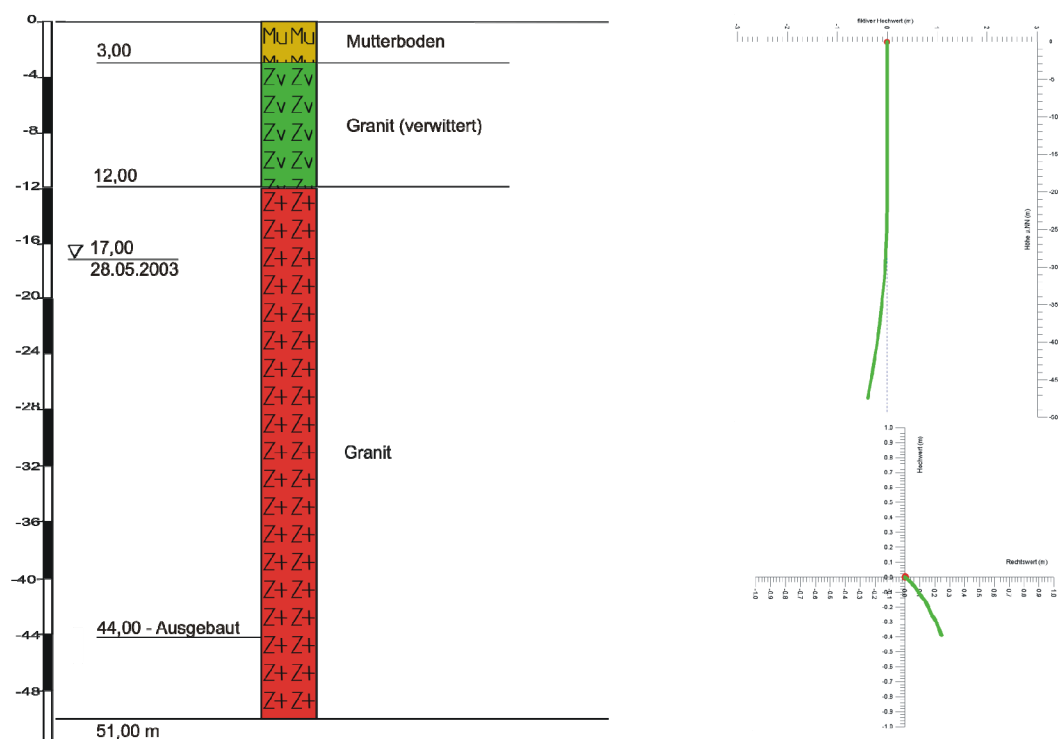


Abb. 2.5: Bohrprofil (links) und Ergebnisse der Abweichung aus der Vertikalen (vertikale Projektion – oben rechts; horizontale Projektion – unten rechts; C & E Engineering & Consulting) für die Station Eiglasdorf.

2 Aufbau und Struktur des Neigungsmesser-Arrays

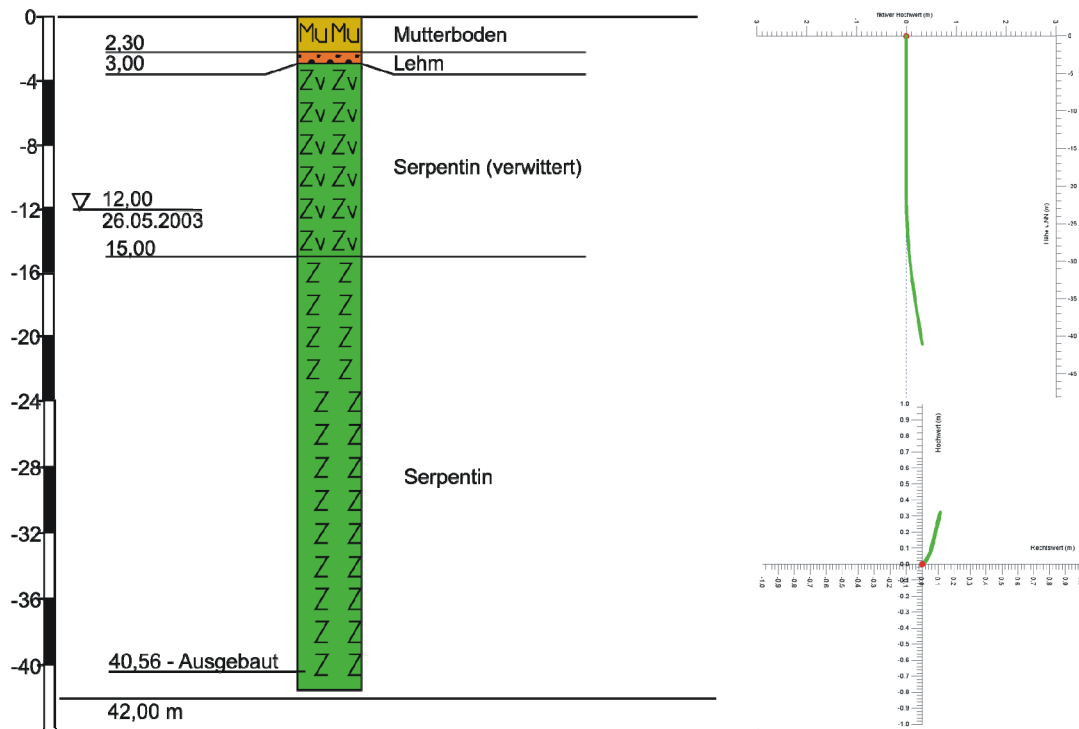


Abb. 2.6: Bohrprofil (links) und Ergebnisse der Abweichung aus der Vertikalen (vertikale Projektion – oben rechts; horizontale Projektion – unten rechts; C & E Engineering & Consulting) für die Station Stockau.

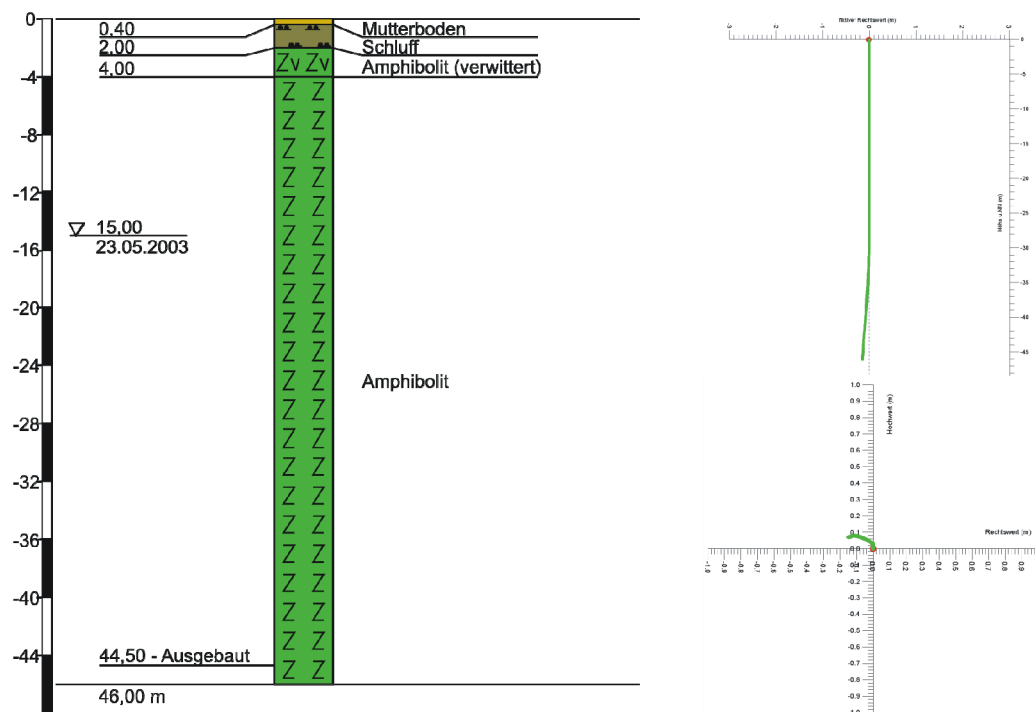


Abb. 2.7: Bohrprofil (links) und Ergebnisse der Abweichung aus der Vertikalen (vertikale Projektion – oben rechts; horizontale Projektion – unten rechts; C & E Engineering & Consulting) für die Station Mittelberg.

2 Aufbau und Struktur des Neigungsmesser-Arrays

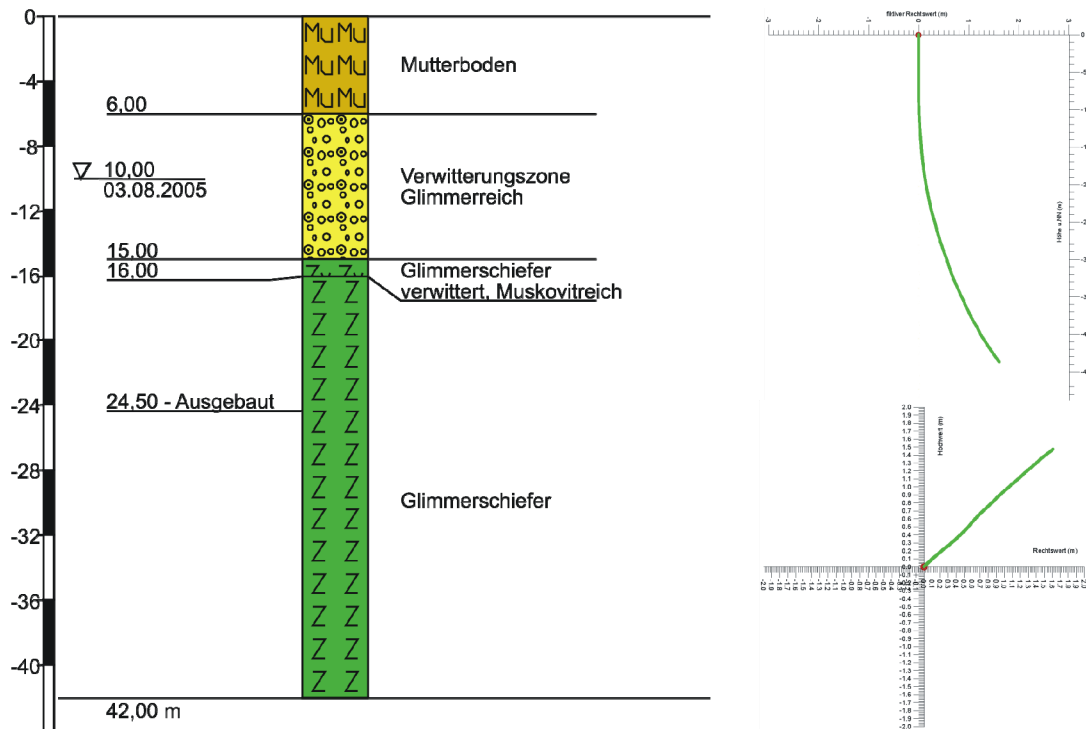


Abb. 2.8: Bohrprofil (links) und Ergebnisse der Abweichung aus der Vertikalen (vertikale Projektion – oben rechts; horizontale Projektion – unten rechts; C & E Engineering & Consulting) für die Station Püllersreuth.

Tab. 2.2: Einbautiefen der Neigungsmesser.

Station	Ausbautiefe in m
Berg	30.5
Eiglasdorf	44.0
Stockau	40.7
Mittelberg	45.5
Püllersreuth	24.5

Tab. 2.3: Parameter der Grundwassermessstellen.

Station	Geographische Koordinaten		Höhe über NN. in m	Ausbautiefe in m
	Rechtswert	Hochwert		
Berg	4510262.7	5519661.1	498.0	30.5
GGA1	4510295.9	5519659.6	497.2	11.5
GGA2	4510282.2	5519677.2	495.9	13.1
Eiglasdorf	4510143.2	5521761.8	506.0	51.0
Stockau	4506755.9	5522456.1	502.8	42.0
GGA1	4506738.3	5522436.9	502.6	11.9
GGA2	4506771.2	5522450.2	501.2	10.7
Mittelberg	4506839.9	5520181.8	532.5	45.0
Püllersreuth	4509119.1	5517297.5	547.3	42.0
GGA1	4509094.1	5517295.8	546.4	10.7
GGA2	4509126.0	5517322.8	546.9	7.9

2.3 Das Funknetzwerk zur Datenübertragung (WLAN)

Das vom GFZ installierte 2,4 GHz Funknetzwerk ist in der folgenden Abb. 2.9 schematisch dargestellt (Asch et al., 2003). Die Übertragung der Daten erfolgt über Richtfunkantennen, welche eine direkte Sichtverbindung benötigen. Bis auf die Station Mittelberg haben alle Stationen eine direkte Anbindung an die KTB. Der Datenstrom der Station Mittelberg wird erst über eine autarke Umsetzerstation umgeleitet, um dann nochmals über Stockau zur KTB geleitet zu werden. Dabei arbeitet die Umsetzerstation nicht die ganze Zeit, sondern wird nur von Zeit zu Zeit aktiviert. Über dieses Funknetzwerk lassen sich die Registriergeräte zudem noch steuern, was eine einfache Wartung und Kontrolle des Netzwerkes ermöglicht. An der KTB wurde zur Sicherung der Daten und zur Steuerung des Netzwerkes ein Server installiert, welcher auch über das Internet bedient werden kann. Weitere Informationen zu den Datenwegen sind unter Kapitel 5.1.7 zu finden.

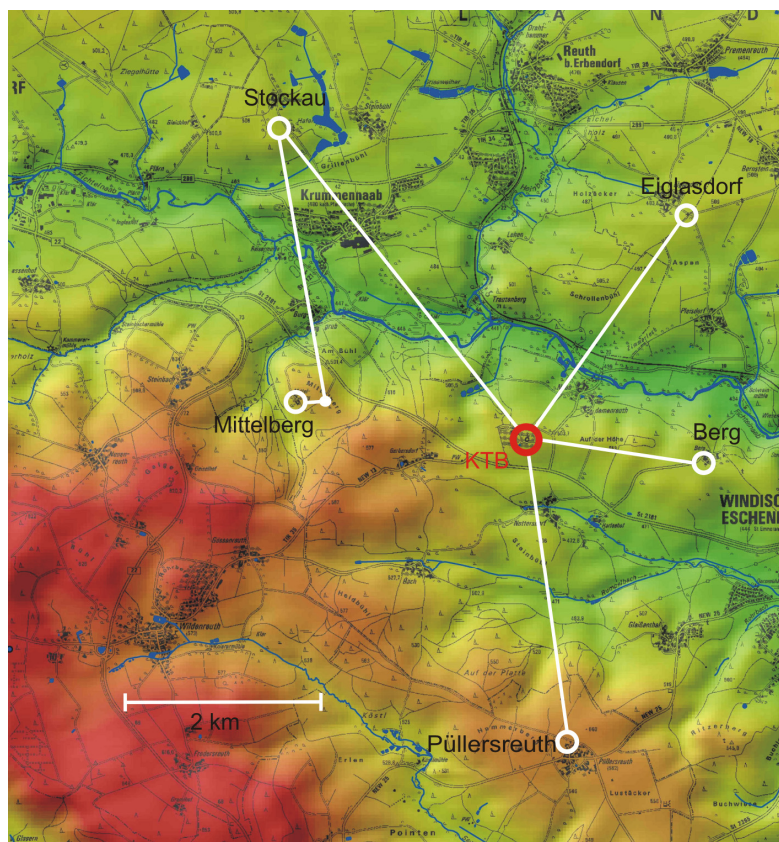


Abb. 2.9: Struktur des Funknetzwerkes (WLAN).

3 Geologie

3.1 Der weite geologische Rahmen

Die Oberfalz, in der sich die KTB befindet, ist ein zentraler Teil des Variskischen Gebirges (Abb. 3.1), welches Europa vom Rand der osteuropäischen Tafel bis zum Atlantik durchzieht. Das nordostbayrische Grundgebirge entstand vor rund 330 bis 280 Millionen Jahren und hatte seinen Entwicklungshöhepunkt gegen Ende des Karbons.

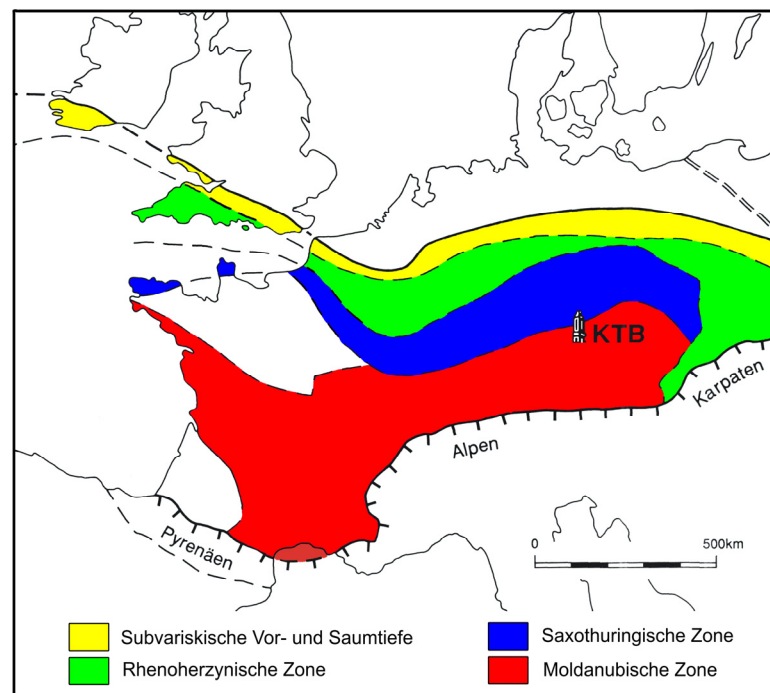


Abb. 3.1: Lokation der KTB im Zonensystem des Variskischen Gebirges (nach Stettner, 1992).

Das Variskische Gebirge unterteilt sich nach Kossmats (1927) in vier Zonen. Nördlich der Alpen befindet sich die Moldanubische Zone, daran schließen sich die

Saxothuringische und Rhenoharzynische Zone an. Den nördlichen Abschluss bildet die Subvariskische Vor- und Saamtiefe.

Das nordbayrische Grundgebirge nimmt südliche Teile der Saxothuringischen Zone mit Frankenwälder Synklinalbereich, dem Münchberger Komplex und dem Antiklinalbereich des Fichtelgebirges ein. Südlich daran schließen sich Teile der Moldanubischen Zone im Oberpfälzer und Bayrischen Wald an. In diesem Teil des Variskischen Gebirges wurde das Rotliegende weitgehend eingeebnet und der Schutt in den Rotliegenden Senken abgelagert. Große Abschnitte wurden in der Folgezeit von mesozoischen bis tertiären Ablagerungen überdeckt. Die orogen verdickte Kruste hat hier eine Mächtigkeit von ca. 30 km.

3.2 Der engere geologische Rahmen

Im Zentrum des KTB-Umfeldes liegt die Neustädter Scholle (Abb. 3.2; Stettner, 1992), welche auch ZEV (Zone von Erbdorf-Vohenstrauß) genannt wird. Die Neustädter Scholle besteht vorwiegend aus Mittel- und Hochdruck-Metamorphiten. Deshalb wird angenommen, dass sie in rund 30 bis 40 km Tiefe subduziert war, bevor sie durch Relativverschiebungen ihre heutige Lage gefunden hat. Die hier anzutreffenden Gesteine sind Gneise, Metabasite und Amphibolite.

Aufgrund des hohen Anteils an Graniten (Abb. 3.2) kann auf sehr große Massenbewegungen sowie eine hohe Durchwärmung der Böhmisches Masse geschlossen werden. In keiner Phase der Erdgeschichte haben hier Granite eine derartig wichtige Rolle gespielt, wie in der jungvariskischen Ära. Nicht nur die Platznahme und die Kristallisation, sondern auch die zugrunde liegende granitoide Mobilisation der tieferen Kruste und des oberen Erdmantels ist Ausdruck eines Vorganges unverhältnismäßiger Wärmeentwicklung. Demzufolge ist der Zeitraum von vor 330 bis 280 Millionen Jahren der bedeutendste wärmegeschichtliche Abschnitt dieses Gebietes.

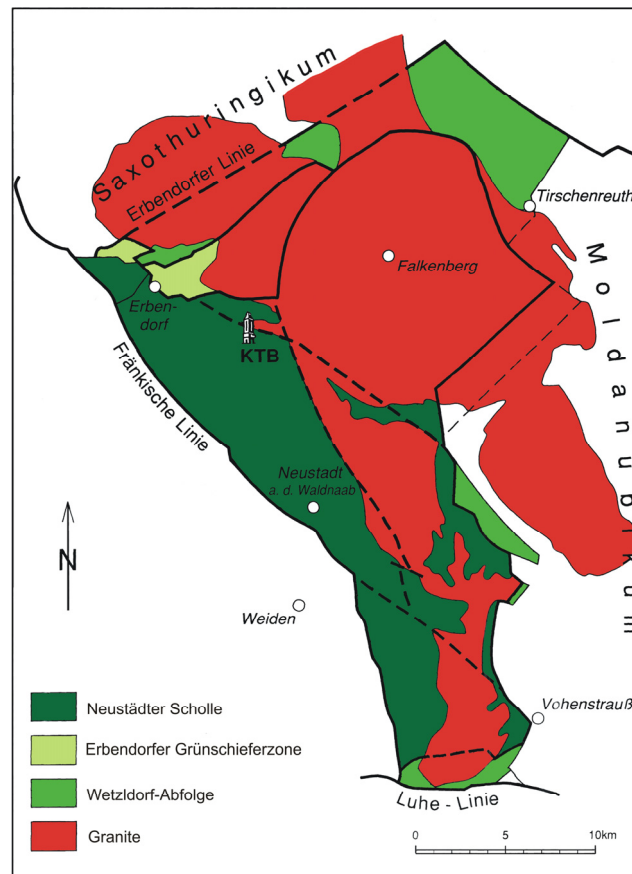


Abb. 3.2: KTB-Umfeld, die Neustädter Scholle, durch Störungszonen in Teilschollen untergliedert, mit den umliegenden geologischen Einheiten (nach Stettner, 1992).

Die Erbdorfer Grünschieferzone besteht aus metamorph überprägten Gesteinen, welche vornehmlich Serpentine sind. Die vierte geologische Einheit, die Wetzldorf-Abfolge, besteht aus altpaläozoischen Folgen. Im Allgemeinen sind dies niedrig metamorphe Gesteine. In der Abfolge sind verschiedene Schiefer, Quarzite und vereinzelte Metebasite zu finden.

Das in Abb. 3.2 dargestellte Gebiet wird von einigen Störungen durchzogen. Als erste ist dabei die Nordsüd – Ostwest verlaufende Fränkische Linie zu nennen. Die Erbdorfer Linie verläuft von Südwest nach Nordost und fällt nach Südosten ein. Im Süden des Gebietes befindet sich die West – Ost ausgerichtete Luhe-Linie.

In Abb. 3.3 ist ein 200 km langer Schnitt zu sehen, welcher unter Zuhilfenahme des seismischen Profils DEKORP 4 (Hirschmann, 1996) erstellt wurde. Das Profil stellt die oben genannten geologischen Einheiten bis zur Mohorovičić-Diskontinuität dar, die in ca. 30 km Tiefe liegt. Die Ausrichtung des Schnittes ist Nordwest – Südost, in der Mitte ist die KTB zu sehen.

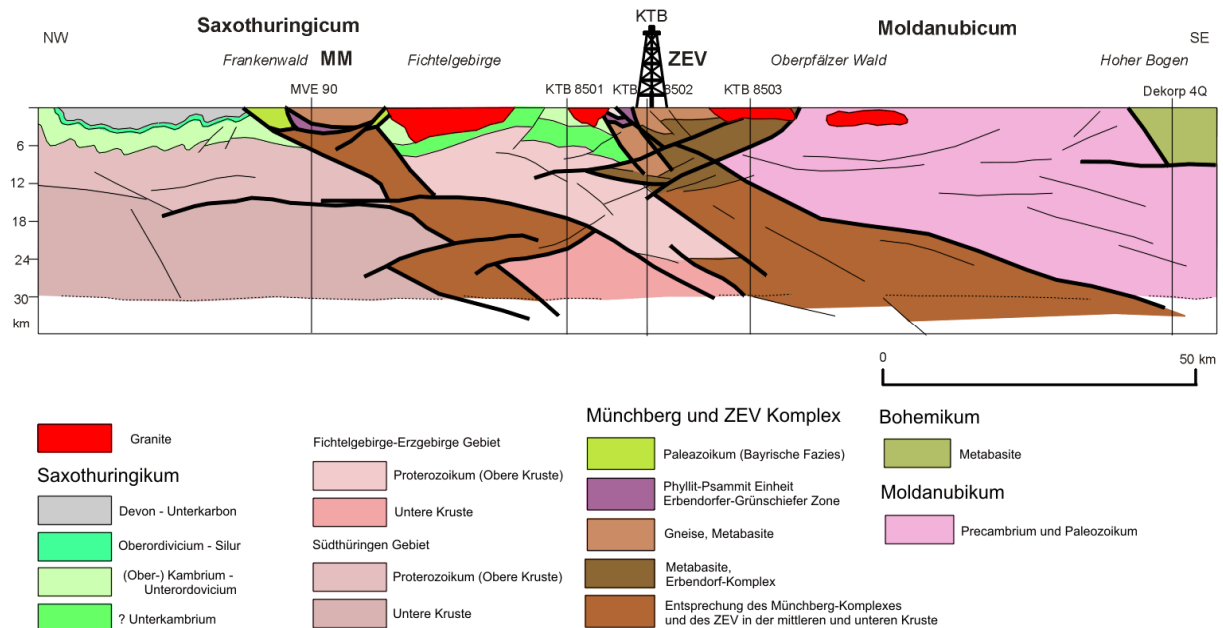


Abb. 3.3: Geologischer Schnitt entlang der Tiefenseismischen Linie DEKORP 4 (nach Hirschmann, 1996).

Da die Stationen einen maximalen Abstand von der KTB von 3.23 km haben, ist auch eine Betrachtung der Geologie in diesem Bereich nötig. Dazu eignet sich das in Abb. 3.4 dargestellte Blockbild vom Umfeld der KTB. Wieder sind die schon in Abb. 3.2 vorgestellten vier geologischen Einheiten und Störungen zu finden. Das System der Störungen ist in Abb. 3.4 detaillierter dargestellt. Das markierte Bohrloch der Hauptbohrung, das bis 9101 m abgeteuft wurde, wird von mehreren Störungen durchzogen. Am wichtigsten sind die mit „SE1“ und „SE2“ markierten. In die von der Hauptbohrung etwa 200 m westlich befindlichen Vorbohrung wird in einer Tiefe von 4 km injiziert. Der Injektionspunkt befindet sich in etwa auf der SE2 Störung.

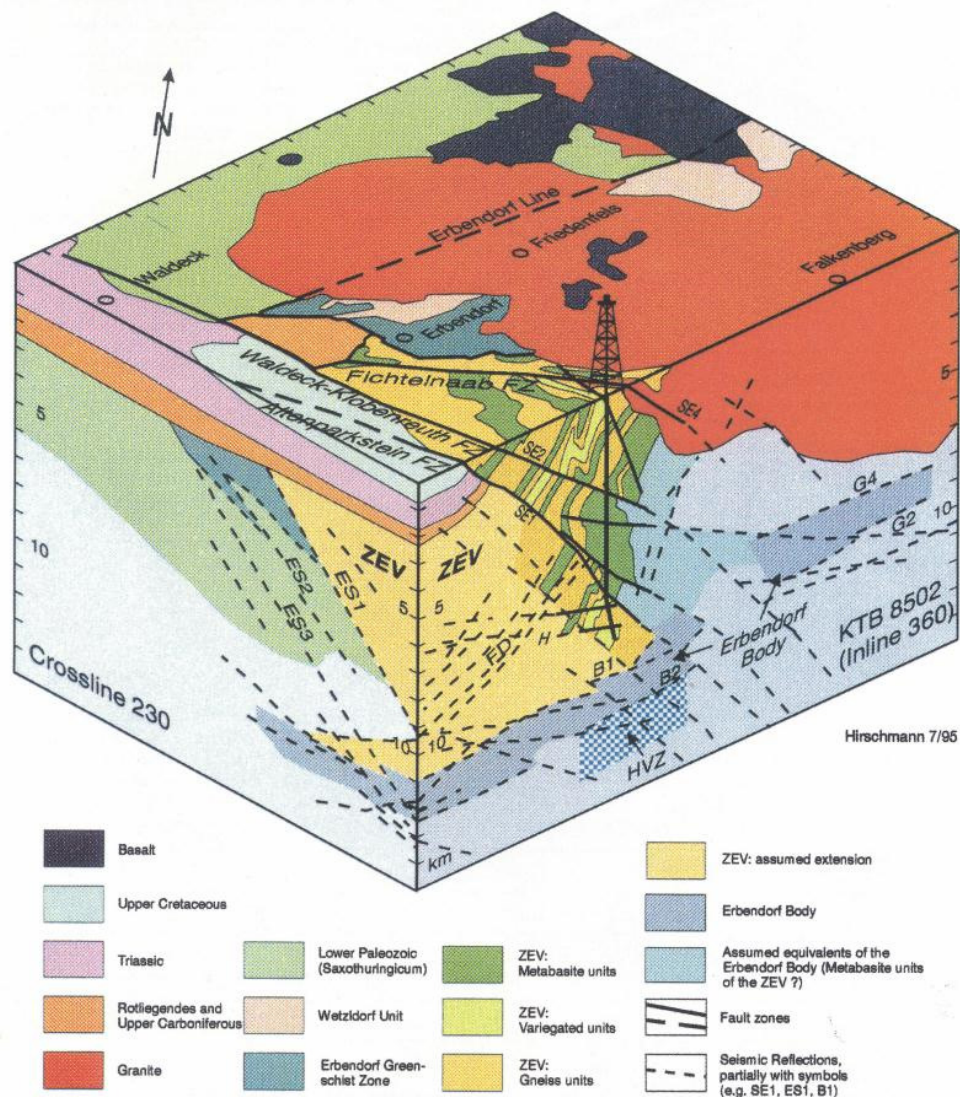


Abb.3.4: Blockbild KTB (Hirschmann, 1995).

3.3 Das geologische Umfeld der KTB mit den Neigungsmesserstationen

Die Geologie der näheren Umgebung (Abb. 3.5) ist stark strukturiert. Die anzutreffenden Gesteine sind größtenteils Magmatite und Granite aber auch vereinzelt jüngere Sedimente, welche allesamt durch eine Schicht Boden bedeckt sind. Die aufgenommenen Bohrprofile (Kapitel 2.2) während der Installationsphase geben die herrschenden Verhältnisse wieder. Bei der Station Püllersreuth ist als anstehendes Gestein in Abb. 2.8 Glimmerschiefer zu finden. Die Station befindet sich laut geologischer Karte im Biotitgneis, in diesem ist vereinzelt Phylonit tektonisch eingeschaltet. Phylonit ist musko-

vitreich, deshalb kann während des Bohrens das angetroffene Material als Glimmerschiefer bezeichnet worden sein.

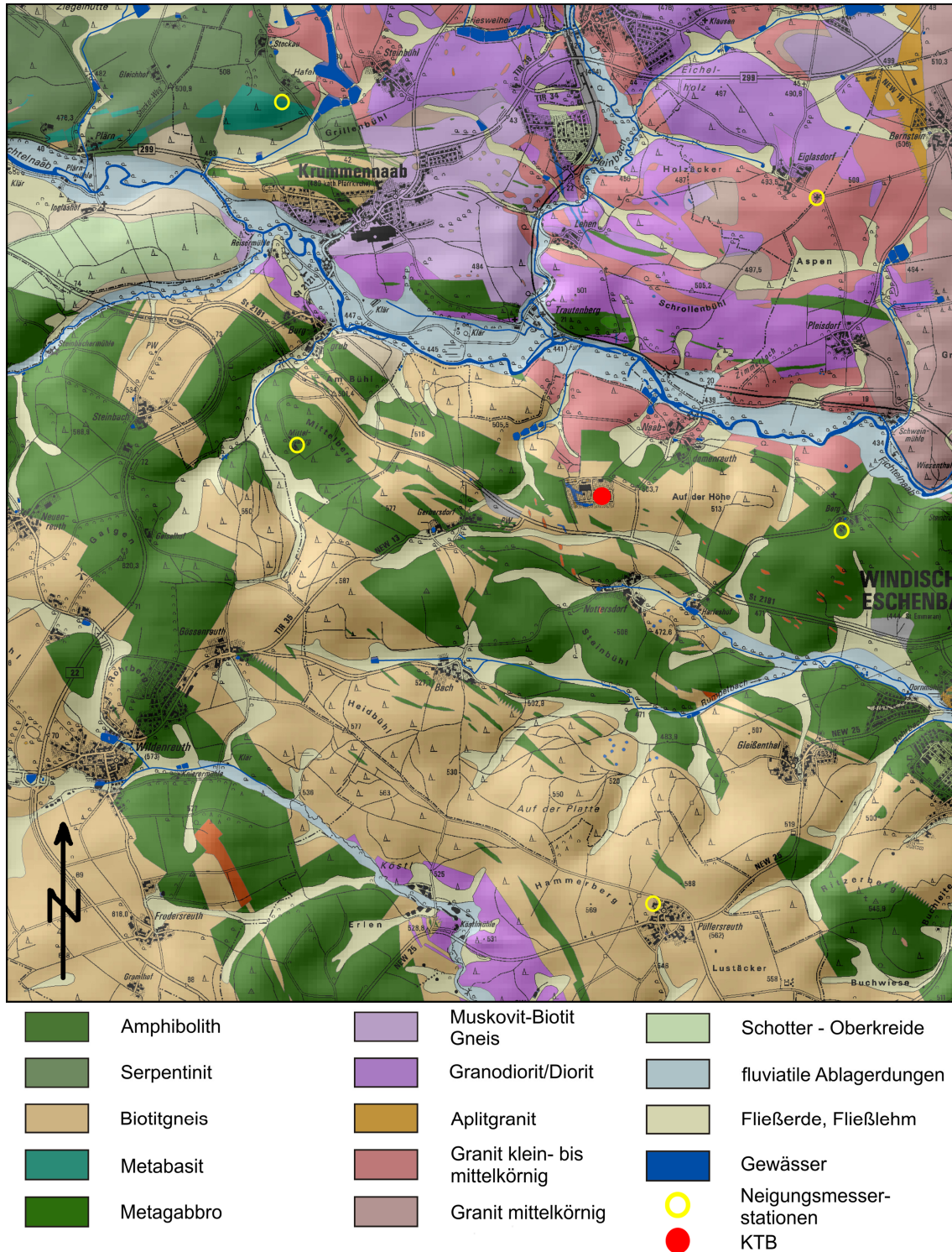


Abb. 3.5: Geologische Karte des Umfeldes der KTB (Bayrisches Geologisches Landesamt).

4 Grundlagen

4.1 Neigungsmesser

Vor dem Jahr 1968 wurden fast ausschließlich Horizontalpendel (Kapitel 5) eingesetzt. Diese hatten den Vorteil der Astasierung¹, aber auch den Nachteil, durch die Aufstellung in Stollen bzw. Kellern dem Cavity Effekt (Harrison, 1976) zu unterliegen. 1968 wurde, durch die Entwicklung moderner Abgriffsysteme der ASKANIA-Werke, der Weg bereitet für den Einsatz von Vertikalpendeln (Kapitel 5). Diese konnten in Bohrlöchern eingesetzt werden, dadurch wird der Cavity Effekt vermieden. Der Nachteil der Bohrlöcher ist, dass diese sehr teuer sind. Möglicherweise ist dies der Grund dafür, dass sich die ASKANIA-Bohrlochneigungsmesser nicht ganz durchsetzen konnten und nur knapp 25 Geräte produziert wurden.

Nach ausführlichen Tests an der TU-Clausthal (Flach & Brauckmann, 1974) wurden die ASKANIA-Bohrlochneigungsmesser in Norwegen eingesetzt. Dabei ging es um die Messung von Krustendeformationen durch wechselnde Auflast eines Stausees (Jentzsch & Koß, 1997). Parallel wurde in Finnland eine Station betrieben, in der Erdbeben im Zusammenhang von ozeanischen und atmosphärischen Auflasten (Weise et al., 1999) studiert wurden. Später wurde ein Neigungsmesser in Japan im Zusammenhang mit der Messung von tektonischen Signalen und ozeanischer Auflast (Ishii et al., 2001) eingesetzt. Von Westerhaus & Zschau (2001) wurden die Instrumente zur Erdbettenforschung in der Türkei eingesetzt.

¹ Astasierung ist die Vergrößerung der mechanischen Empfindlichkeit durch eine entsprechende Konstruktion. Dadurch wird die Dynamik des Messgerätes erhöht, weil das gemessene Signal nicht linear proportional zum Eingangssignal ist (Torge, 1989).

4.2 Signalgehalt von Neigungsmessungen

Bei der Auswertung einer geophysikalischen Zeitreihe ist der Signalgehalt zu betrachten. Die registrierte Zeitreihe enthält neben dem gesuchten Anteil der Messgröße auch Rauschen und einen instrumentell erzeugten Anteil. In Abb. 4.1 sind die Signalanteile schematisch dargestellt.

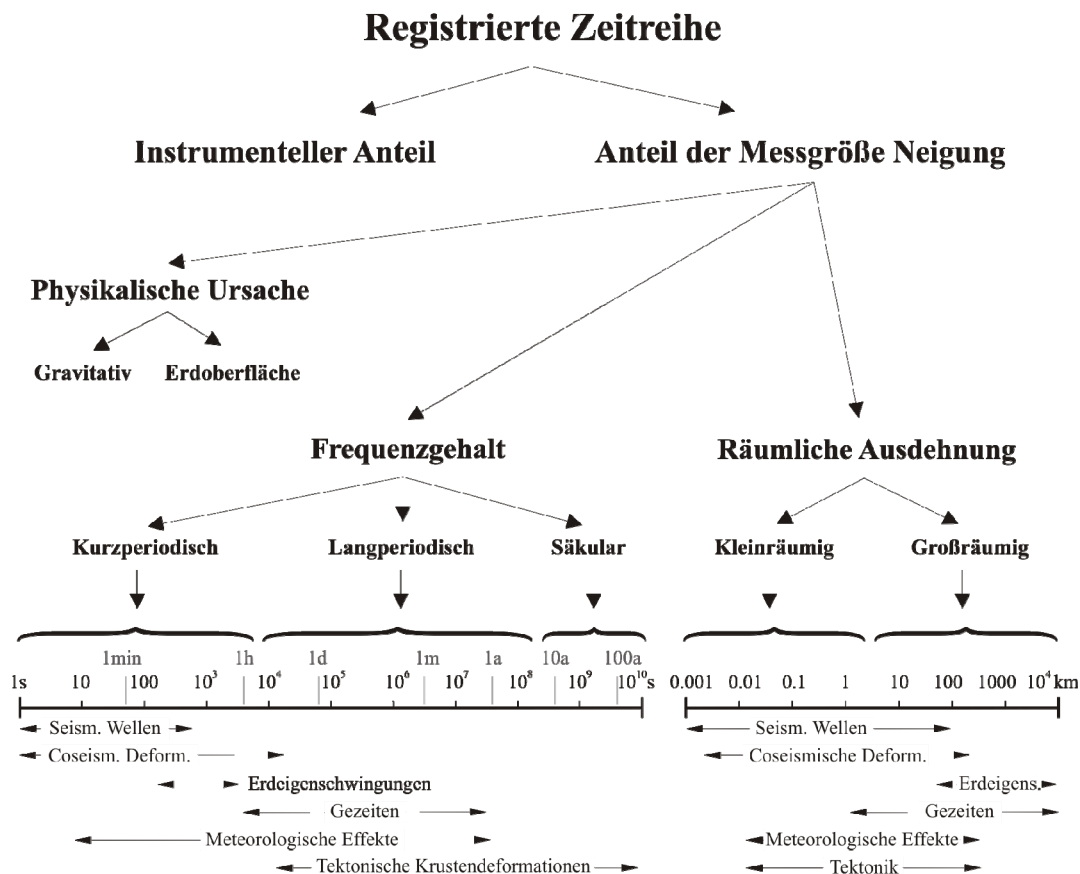


Abb. 4.1: Signalgehalt von Neigungsmessungen und mögliche Klassifikation (Kämpel, 1982); die physikalische Ursache ist in horizontale Beschleunigung (gravitativ) und Deformation der Erdoberfläche unterteilt.

An dieser Stelle soll nur auf den Signalanteil der echten Neigungen eingegangen werden. Der instrumentelle Anteil der Zeitreihe wird in Kapitel 5 näher betrachtet. Der Gehalt der Signale kann nach Kämpel (1982) klassifiziert werden nach:

- der physikalischen Ursache (Attraktion, Deformation)
- dem Frequenzgehalt (im Zeit-Raum: kurz-, langperiodisch, säkular)
- der Wellenlänge (im Orts-Raum: klein-, großräumig)

Physikalische Ursachen

Die physikalischen Ursachen setzen sich aus zwei Anteilen zusammen. Neigungen können durch gravitative Attraktion (Abb. 4.2) und durch Deformation der Erdoberfläche (Abb. 4.3) erzeugt werden. Die Trennung der Ursachen kann nur durch eine numerische Modellierung erfolgen.

Der Schwerevektor \vec{g} zeigt in erster Näherung auf das Massenzentrum der Erde und steht lotrecht auf der Äquipotentialfläche (Abb. 4.2).

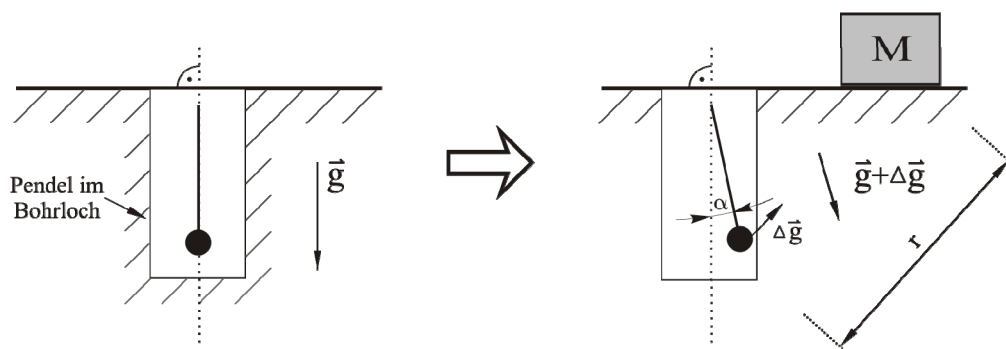


Abb. 4.2: Neigung durch Attraktion (nach Kümpel, 1982).

Wird eine genügend große Masse M in ausreichender Nähe r zum Neigungsmesser gebracht, so erfolgt mit $\Delta \vec{g} = M / r^2$ eine Auslenkung des Schwerevektors und somit auch die des Sensors um den Winkel α .

Bei Neigungen durch Deformation ändert sich der Schwerevektor \vec{g} nicht (Abb. 4.3). Das Neigungssignal wird durch die Deformation der Erdoberfläche und damit einer Drehung der mit der Erdoberfläche verbundenen Verrohrung gegenüber dem Sensor um den Winkel α produziert.

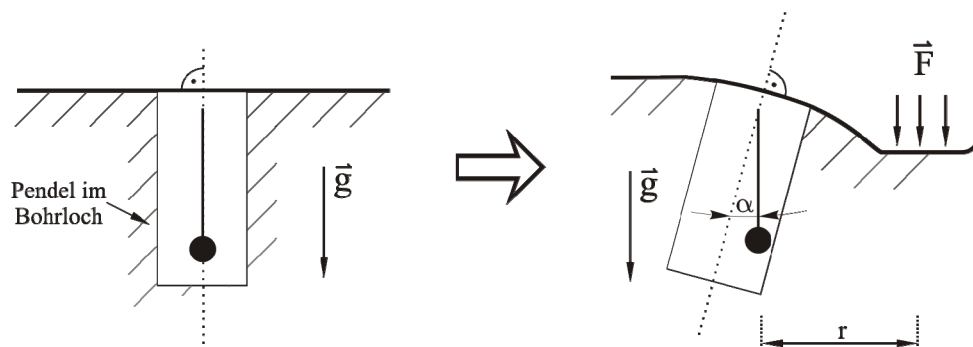


Abb. 4.3: Neigung durch Deformation (nach Kämpel, 1982).

Frequenzgehalt

Geophysikalische Vorgänge wie Neigungsänderungen bestehen aus einer Überlagerung von Attraktion und Deformation. Deshalb ist es sinnvoll, die Neigungssignale nach ihrem Spektralgehalt im Zeit-Raum oder im Orts-Raum zu beschreiben. Die Betrachtung des Frequenzgehaltes im Zeit-Raum erweist sich dabei als vorteilhaft. Dabei lässt sich durch den Frequenzinhalt in etwa die Ursache (Grundwasser, Luftdruck, temperaturinduzierte Effekte und tektonische Krustendeformationen) des Signals ermitteln. Gezeitenbedingte Neigungen unterscheiden sich in einem Aspekt grundlegend von allen anderen neigungsverursachenden Effekten. Während Tektonik, Seismologie, Grundwasser, Temperatur und Luftdruck sporadische und aperiodische Neigungen auslösen, üben Gezeiten (Kapitel 4.2) eine streng harmonische Anregung aus, deren Perioden besonders gut bekannt sind.

4.3 Gezeitenpotential

Die Gezeitenbeschleunigung \vec{b} an einem beliebigen Punkt P der Erde mit der Zenitdistanz ϑ ergibt sich aus der Summe der Gravitationsbeschleunigung \vec{b}_p , die der betrachtete Himmelskörper auf den Punkt P ausübt, und der Bahnbeschleunigung $-\vec{b}_0$, die von der Bewegung der Erde um den gemeinsamen Schwerpunkt des Zweikörpersystems herrührt (in Abb. 4.4; nach Wenzel, 1997a).

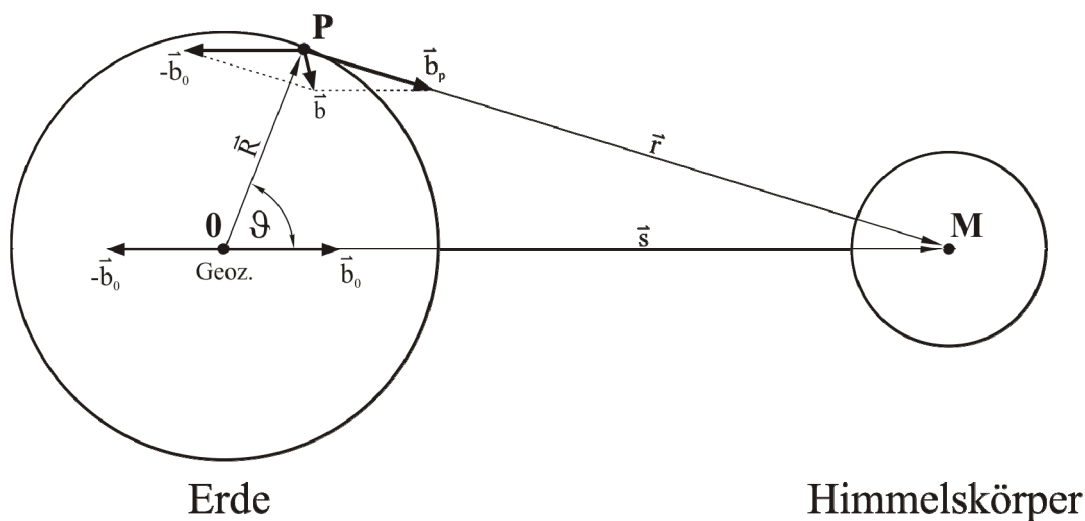


Abb. 4.4: Ableitung der Gezeitenbeschleunigung (nach Wenzel, 1997a).

Die Gezeitenbeschleunigung resultiert nach Newtons Gravitationsgesetz aus

$$\vec{b} = \vec{b}_p - \vec{b}_0 = \frac{\gamma M}{r^2} \cdot \frac{\vec{r}}{r} - \frac{\gamma M}{s^2} \cdot \frac{\vec{s}}{s} \quad (4.1)$$

wobei M die Masse des Himmelskörpers ist. Der Abstand zwischen den beiden Schwerpunkten der Körper wird mit \vec{s} bezeichnet und die Abstände der Schwerpunkte zu dem betrachteten Punkt P mit \vec{R} und \vec{r} . Die Bahnbeschleunigung $-\vec{b}_0$ kann dabei in guter Näherung als die negative Gravitationsbeschleunigung \vec{b}_0 des Himmelskörpers auf das Geozentrum betrachtet werden.

Mit $\vec{b} = \text{grad}V = \partial V / \partial \vec{r}$ führt man das skalare Gezeitenpotential V der vektoriellen Gezeitenbeschleunigung \vec{b} ein, um daraus die Kugelfunktionsentwicklung und so die spektrale Zerlegung des Gezeitenpotentials in längen- und breitenabhängige Terme abzuleiten. Um die Zusatzbedingung von $V = 0$ im Geozentrum zu erfüllen, muss vom Gesamtpotential ein Term $\gamma M / s$ abgezogen werden. Unter Verwendung eines rechtwinkligen, nichtrotierenden Koordinatensystems im Punkt O lässt sich das Potential von \vec{b}_0 durch $|\vec{b}_0| \cdot R \cos \vartheta$ ausdrücken, woraus das Gezeitenpotential folgt:

$$V = \gamma M \left(\frac{1}{r} - \frac{1}{s} - \frac{R \cos \vartheta}{s^2} \right). \quad (4.2)$$

Wird im Dreieck POM (Abb. 4.4) der Kosinussatz $r^2 = s^2 + R^2 - 2sR \cos \vartheta$ angewendet und nach $1/r$ aufgelöst, so kann man den Quotienten $1/r$ in eine Taylor-Reihe nach Potenzen von R/s entwickeln. Das Ergebnis dieser Entwicklung führt auf die Legendre'schen Polynome $P_n(\cos \vartheta)$. Das Ersetzen von $1/r$ in der Gleichung (4.2) durch diese Polynome ergibt das Gezeitenpotential V in der Form:

$$V = \frac{\gamma M}{s} \cdot \sum_{n=2}^{\infty} \left(\frac{R}{s} \right)^n P_n(\cos \vartheta) \quad (4.3)$$

Diese Gleichung (4.3) beschreibt das Gezeitenpotential eines Himmelskörpers. Zur Berechnung der Gesamtwirkung muss dieses Potential für alle zu betrachtenden Körper aufgestellt werden. Das Verhältnis R/s von Erdradius zur Entfernung vom Himmelskörper beträgt für den Mond $1.6 \cdot 10^{-2}$ und für die Sonne $4 \cdot 10^{-5}$. Die Reihenentwicklung konvergiert demzufolge sehr schnell. Sie wird heute bis maximal $n = 6$ für den Mond betrieben, für die Sonne bis $n = 3$ und für die Planeten bis $n = 2$. Mit ca. 98% des gesamt-

ten Gezeitenpotentials liefert das Potential V_2 bereits für $n = 2$ den größten Beitrag (Wenzel, 1997a).

Doodson (1922) führte für den Mond die Gezeitenkonstante G

$$G(R) = \frac{3}{4} \gamma M \frac{R^2}{c^3} \quad (4.4)$$

ein, wodurch sich das Gezeitenpotential (4.3) mit der mittleren Entfernung von Erde zu Mond c und unter Beschränkung auf die Hauptterme V_2 und V_3 ausdrücken lässt als:

$$\begin{aligned} V &= G(R) \left\{ \left[\left(\frac{c}{s} \right)^3 \cdot \left(\cos 2\vartheta + \frac{1}{3} \right) \right] + \left[\frac{1}{6} \frac{R}{c} \left(\frac{c}{s} \right)^4 \cdot (5 \cos 3\vartheta + 3 \cos \vartheta) \right] + \dots \right\} \\ &= V_2 + V_3 + \dots \end{aligned} \quad (4.5)$$

Gleiches gilt mit den entsprechenden Größen für die Sonne, wobei der Wert der Gezeitenkonstante G_S der Sonne bei nur etwa 46% des Wertes des Mondes G liegt.

Interessiert man sich für die Wirkung des Gezeitenpotentials V auf die Erde, fragt man also nach den Gezeitenbeschleunigungen Z in radialer und H in horizontaler Richtung, muss man die Gleichung (4.5) partiell differenzieren. Für V_2 gilt (Bartels, 1957):

$$\begin{aligned} Z &= \left(\frac{\partial V_2}{\partial R} \right)_{c=s} = \frac{2G(R)}{R} \cdot \left(\cos 2\vartheta + \frac{1}{3} \right) \\ H &= - \left(\frac{\partial V_2}{R \partial \vartheta} \right)_{c=s} = \frac{2G(R)}{R} \cdot \sin 2\vartheta \end{aligned} \quad (4.6)$$

Die tatsächlichen Wirkungen der Gezeitenbeschleunigung auf die reale elastische, nicht „starre“ Erde weichen von den hier berechneten theoretischen Werten ab. Der in der Geodynamik wichtige Quotient von beobachteten zu theoretischen Gezeiten (genannt γ -Faktor bei Neigungen und δ -Faktor für die Schwere) quantifiziert diese Abweichung und beträgt für Neigungen in der Regel etwa 0.7; der genaue Wert ist abhängig vom Erdmodell.

Die bisherige Betrachtung des Gezeitenpotentials (4.5) im nichtrotierenden Koordinatensystem setzt stets die Kenntnis der momentanen geozentrischen Zenitdistanz ϑ zu Mond oder Sonne voraus, um das Gezeitenpotential für einen Punkt auf der Erdoberfläche zu berechnen. Dieser Ansatz ist in der Praxis unhandlich. Viel besser ist es, für den Beobachtungspunkt zum konventionellen mitrotierenden Koordinatensystem von Länge, Breite und Abstand vom Erdzentrum überzugehen. In der üblichen Projektion vom Erdmittelpunkt auf die Himmelskugel seien (Abb. 4.5) C der Himmelspol, M der Mond (Deklination δ , Zenitdistanz ϑ), P der Beobachtungsort (geozentrische Brei-

te φ) und CP sein Meridian. Wahre Mondzeit τ_1 sei von der unteren Kulmination gerechnet in Grad, so dass der Stundenwinkel des Mondes gleich $(\tau_1 - 180^\circ)$ ist. Dann ist im Dreieck CMP :

$$\cos \vartheta = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos (\tau_1 - 180^\circ) \quad (4.7)$$

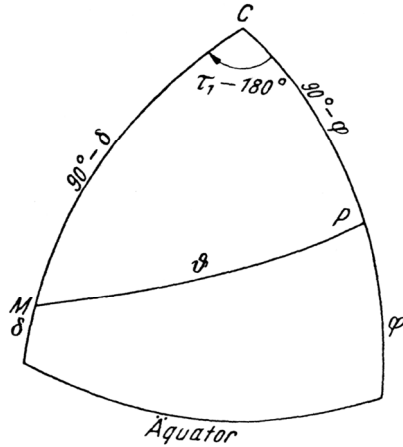


Abb. 4.5: Himmelsdreieck (aus Bartels, 1957).

Mit der Ersetzung von $\cos \vartheta$ (4.7) in den Haupttermen V_2 und V_3 (4.7) und der Verwendung von $\cos 2\vartheta = 2\cos^2 \vartheta - 1$ und $\cos 3\vartheta = 4\cos^3 \vartheta - 3\cos \vartheta$ ergibt sich:

$$V_2 = G(R) \cdot \left(\frac{c}{s} \right)^3 \cdot \left[3 \left(\frac{1}{3} - \sin^2 \delta \right) \left(\frac{1}{3} - \sin^2 \varphi \right) - \sin 2\varphi \sin 2\delta \cos \tau_1 + \cos^2 \varphi \cos^2 \delta \cos 2\tau_1 \right] \quad (4.8)$$

$$V_3 = G(R) \cdot \frac{R}{c} \cdot \left(\frac{c}{s} \right)^4 \cdot \left[\frac{1}{3} (3 \sin \varphi - 5 \sin^3 \varphi) (3 \sin \delta - 5 \sin^3 \delta) - \frac{1}{2} \cos \varphi (1 - 5 \sin^2 \varphi) \cos \delta (1 - 5 \sin^2 \delta) \cos \tau_1 + 5 \sin \varphi \cos^2 \varphi \sin \delta \cos^2 \delta \cos 2\tau_1 - \frac{5}{6} \cos^3 \varphi \cos^3 \delta \cos 3\tau_1 \right] \quad (4.9)$$

Damit folgen für das Gezeitenpotential $V = V_2 + V_3 + \dots$ die jeweiligen Perioden, welche sich direkt aus den Haupttermen (4.8),(4.9) zeilenweise ergeben:

18,6 Jahre	langperiodische Wellen	(zonale KF)
Jahr, Halbjahr	langperiodische Wellen	(zonale KF)
~28 und ~14 Tage	langperiodische Wellen	(zonale KF)
~24 Stunden	tägliche Wellen	(tesserales KF)
~12 Stunden	halbtägige Wellen	(sektorielle KF)
~8 Stunden	dritteltägige Wellen	(sektorielle KF)
~6 Stunden	vierteltägige Wellen	(sektorielle KF)

Die vierteltägigen Wellen ergeben sich aus V_4 .

Alle notwendigen Informationen zum spektralen Gezeitenpotential werden in so genannten Gezeitenkatalogen aufgelistet, deren Genauigkeit sich ständig verbessert. Diese Entwicklung wird möglich durch fortschreitend genauere Bestimmung der astronomischen Parameter und die wachsenden Rechenleistungen und ist nötig wegen der steigenden Auflösung von Gravimetern (heute $0,01 \cdot 10^{-9} \text{ m/s}^2$). Die im Gezeitenpotential V verwendeten Sinus- und Kosinus-Terme mit konstanter Frequenz und Amplitude werden als „Wellen“ bezeichnet. In den Gezeitenkatalogen sind die einzelnen Wellen nach ihren jeweiligen Frequenzen angeordnet und mit einer Buchstaben/Zahlenkennung beschriftet, die aus der Reihenentwicklung von Doodson herrührt.

Die Amplituden der Gezeitenbeschleunigungen hängen nur von der geozentrischen Breite des Beobachtungsortes und seiner Entfernung vom Erdschwerpunkt ab. Damit die beiden Komponenten der Gezeitenbeschleunigung Z in radialer und H in horizontaler Richtung berechnet werden können, wird das spektrale Gezeitenpotential V (4.6) partiell abgeleitet. Am Beobachtungsort der Länge λ und Breite φ kann man dabei die Horizontalkomponente H der Gezeitenbeschleunigung in ihre beiden Bestandteile Y in Nord-Süd Richtung und X in Ost-West Richtung zerlegen (Melchior, 1983):

$$\begin{aligned} Y &= \frac{\partial V_s}{R \partial \varphi} \\ X &= \frac{\partial V_s}{R \cos \varphi \cdot \partial \lambda} \end{aligned} \quad (4.10)$$

Sieht man die Entfernung einer jeden Station zum Erdschwerpunkt in erster Näherung als konstant an, lässt sich die Gezeitenneigung als Ausdruck der wirkenden Gezeitenbeschleunigungen X und Y als Funktion der Breite darstellen (Abb. 4.6).

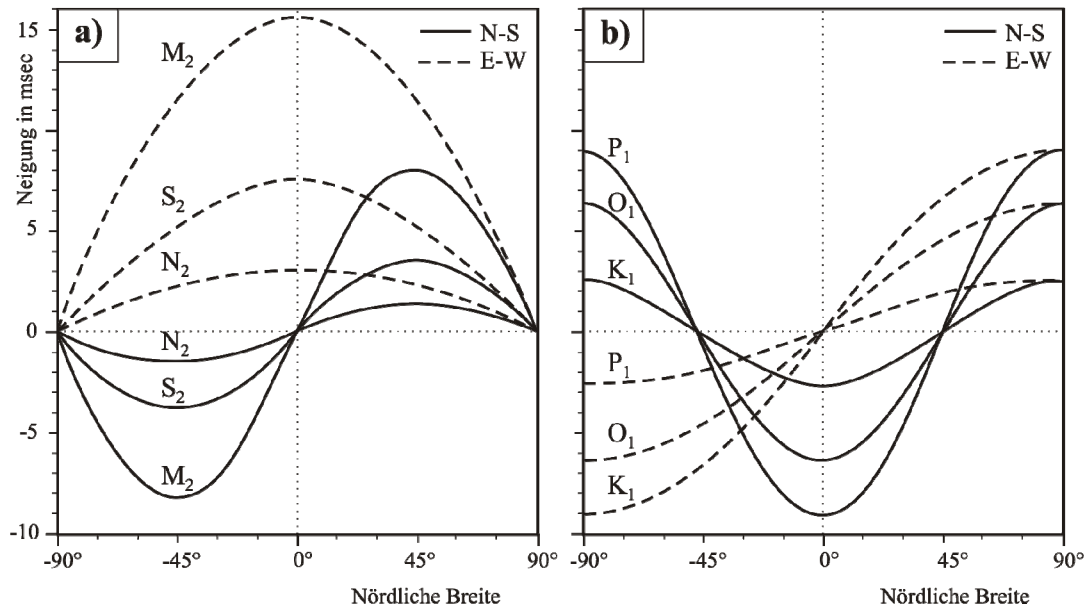


Abb. 4.6: Amplituden der Gezeitenneigung durch die horizontale Gezeitenbeschleunigung für die wichtigsten halbtägigen (a) und ganztägigen (b) Wellen als Funktion der Stationsbreite (nach Melchior, 1983).

4.4 Poroelastische Rheologie und punktförmige Fluidinjektion

Zur Beschreibung der physikalischen Prozesse bei einer punktförmigen Fluidinjektion wird die Theorie der poroelastischen Rheologie herangezogen, welche die Verknüpfung des Hooke'schen Gesetzes für 3D-Körper mit dem Darcy-Gesetz ist. Die folgenden Ausführungen basieren auf dem Vorlesungsskript „Poroelastische Rheologie“ von Kümpel (2005).

Dazu sind einige Annahmen und Voraussetzungen nötig, welche in der Tab. 4.1 gegeben sind:

Tab. 4.1: Annahmen und Voraussetzungen der poroelastischen Theorie (nach Kümpel, 2005).

Annahme/ Voraussetzung	Erklärung
Medium als Kontinuum	keine Auflösung der Porenstruktur
Gültigkeit des Hooke'schen Gesetzes	kleine Kräfte und Verformungen
Gültigkeit des Darcy-Gesetzes	nur laminares Fließen
Materialparameter sind konstant	unabhängig von Druck und Verformung
Prozess ist isothermal	keine Temperaturänderung
Medium ist chemisch inert	Material bleibt unverändert
Medium ist isotrop	keine Richtungsabhängigkeit
Medium ist gesättigt	nur geringe Mengen Porengas

Rheologisches Verhalten lässt sich mit Hilfe von Ersatzschaltbildern darstellen. Abb. 4.7 zeigt diese für die poroelastische Theorie enthalten. Dabei werden einfache mechanische Elemente verwendet und gegebenenfalls gekoppelt. Der Biot-Körper ist das Model, welches die physikalisch wichtigen Parameter veranschaulicht.

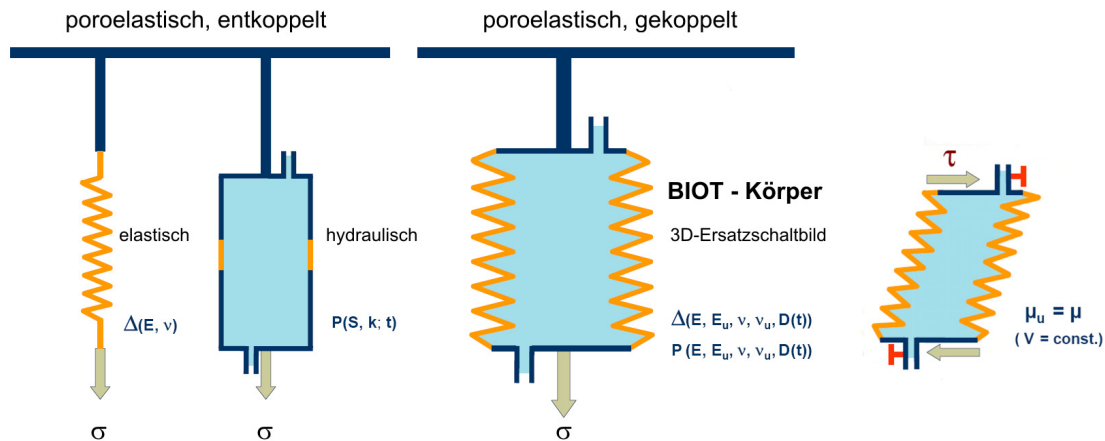


Abb. 4.7: Ersatzschaltbilder für die Rheologie für poroelastisches Verhalten (nach Kümpel, 2005).

Das Hooke'sche Gesetz wird mit Hilfe einer elastischen Feder veranschaulicht. Dabei ist die Deformation Δ bei einer definierten Spannung σ von den Elastizitätsparametern E und ν abhängig. Der hydraulische Anteil, welcher aus dem Darcy-Gesetz hervorgeht, lässt sich mit Hilfe einer geschlossenen Röhre mit Zu- bzw. Abfluss versehen symbolisieren. Weiterhin lässt sich das Volumen dieser Röhre, beispielsweise über eine Schubmanschette ändern. Der Porendruck bei wiederum definierter Spannung ist vom S (Speicherkoeffizient), k und t abhängig. Der beschriebene entkoppelte Fall (Abb. 4.7, Mitte) lässt sich gedanklich nutzen, um sich an die Lösung der Differentialgleichungen iterativ heranzutasten.

Der Biot-Körper (Abb. 4.7, rechts) ist die Kopplung des elastischen und hydraulischen Elements. Die Deformation Δ und der Druck P sind bei definierter Spannung σ abhängig von den Elastizitätsparametern E, E_u, ν und ν_u , und von der hydraulischen Diffusivität $D(t)$. Die Besonderheit des Biot-Körpers stellt sich in der Unabhängigkeit des Schermoduls von drainierten und undrainierten Bedingungen dar (Abb. 4.7, links). Die Bedingungen lassen sich in den Ersatzschltbildern durch öffnen bzw. schließen des Zu- und Abflusses realisieren. Für die jeweiligen Bedingungen sind unter-

schiedliche Elastizitätsparameter anzuwenden. Im drainiertem Fall heißen diese E und ν , im Undrainiertem E_u und ν_u .

Eine Liste der verwendeten Variablen und Symbole ist im Anhang I zu finden. Das Hooke'sche Gesetz für einen 3D-Körper lautet:

$$E e_{ij} = (1 + \nu) \sigma_{ij} + 2\nu P_c \delta_{ij} \quad (4.11)$$

mit

$$P_c = -(\sigma_{11} + \sigma_{22} + \sigma_{33})/3 \quad (4.12)$$

Die Variablen E und ν sind die Elastizitätsparameter des kompakten Körpers und σ_{ij} beschreiben die herrschenden Spannungen. Die Verformungen e_{ij} sind Ableitungen der Verschiebungen u_{ij} und lassen sich wie folgt darstellen:

$$e_{ij} = \frac{\partial u_i}{\partial x_j} \quad (4.13)$$

$$e_{ij} = \frac{\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}}{2} \quad (4.14)$$

$$\sum_{j=1}^3 \frac{\partial \sigma_{ij}}{\partial x_j} = 0 \quad (4.15)$$

Mit den Gleichgewichtsbedingungen (4.15), wobei die Summe der Spannungsdifferenzen gleichgerichteter Spannungen null sind, ergeben sich die Bewegungsgleichungen (4.16) und die Spannungs-Dehnungs-Beziehungen (4.22), (4.23) für poröse, gesättigte Körper. Die Körperkräfte werden durch $\alpha \frac{\partial P}{\partial x_i}$ beschrieben.

$$\mu \nabla^2 u_i + \frac{\mu}{1-2\nu} \frac{\partial \Delta_v}{\partial x_i} = \alpha \frac{\partial P}{\partial x_i} \quad (4.16)$$

mit

$$\mu = \frac{E}{2(1+\nu)} \quad (4.17)$$

$$\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} \quad (4.18)$$

$$\Delta_v = e_{11} + e_{22} + e_{33} \quad (4.19)$$

Durch die Körperkräfte ist der Druck P eine neue Variable, wodurch die Gleichung (4.20) erforderlich ist. Der Ausdruck $\frac{1}{Q}$ ist der Biot'sche Kompressibilitätsparameter (4.21). Die Gleichungen (4.22) und (4.23) beschreiben die Spannungs-Dehnungs-Beziehungen für die poroelastische Rheologie.

$$\frac{1}{Q} \frac{\partial P}{\partial t} + \alpha \frac{\partial \Delta_v}{\partial t} = \kappa \nabla^2 P \quad (4.20)$$

$$\frac{1}{Q} = \frac{9}{2} \frac{(1-2\nu)(\nu_u - \nu)}{2(1-2\nu)(1+\nu_u)^2 \mu B^2} = S - \alpha^2 c \quad (4.21)$$

$$E e_{ij} = (1+\nu) \sigma_{ij} + \left[2\nu P_c + \frac{3(\nu_u - \nu)}{(1+\nu_u)B} \right] \delta_{ij} \quad (4.22)$$

$$\frac{9(\nu_u - \nu)}{(1+\nu_u)EB} \frac{\partial \left(\frac{P}{B} - P_c \right)}{\partial t} = \kappa \nabla^2 P \quad (4.23)$$

Wie mit verschiedenen Messgeräten an der Erdoberfläche die Deformationskomponenten gemessen werden können, ist in Abb. 4.8. dargestellt.

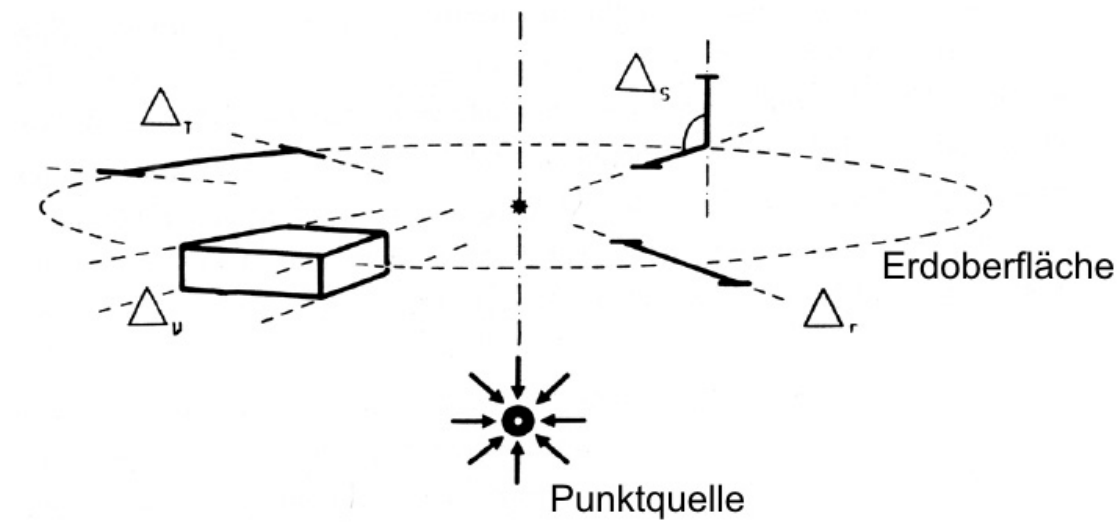


Abb. 4.8: Geometrien von Volumen-, Linien- und Winkелеlementen zur Definition von Dilatation Δ_v und Radial-, Tangential- und Scherverformung Δ_r , Δ_T , Δ_s in der Umgebung von Punktquellen (nach Kämpel, 1989).

Die einzelnen Deformationskomponenten in Zylinderkoordinaten lauten:

$$\Delta_V = e_{rr} + e_{\theta\theta} + e_{zz} = \frac{\partial u_r}{\partial r} + \frac{1}{r} \left(\frac{\partial u_\theta}{\partial \theta} + u_r \right) + \frac{\partial u_z}{\partial z} \quad (4.24)$$

$$\Delta_r = e_{rr} = \frac{\partial u_r}{\partial r} \quad (4.25)$$

$$\Delta_T = e_{\theta\theta} = \frac{1}{r} \left(\frac{\partial u_\theta}{\partial \theta} + u_r \right) \quad (4.26)$$

$$\Delta_S = e_{rz} = \frac{1}{2} \left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} \right) \quad (4.27)$$

Und bei Axialsymmetrie ergibt sich:

$$\Delta_V = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} \quad (4.28)$$

$$\Delta_T = \frac{u_r}{r} \quad (4.29)$$

weil $\frac{\partial u_\theta}{\partial \theta} = 0$.

Als Voraussetzung für die folgenden Gleichungen, die Fundamentallösungen für anhaltende Porendruckstörungen wird zum Anfangszeitpunkt der Gleichgewichtszustand erwartet. Das bedeutet, dass jegliche früheren Einflüsse abgeklungen sind. Als Anregung wird eine Punktquelle angenommen, in der mit konstanter Rate Q_f Fluid injiziert wird. Die Fluidvolumenrate kann mittels der folgenden Gleichungen (4.30) ab dem Zeitpunkt $t = 0$ angegeben werden.

$$\begin{aligned} Q &= V_f / t \\ &= m_f (\rho_f t) \end{aligned} \quad (4.30)$$

Der Druck P lässt sich über Gleichung (4.31) und die Spannung über Gleichung (4.32) berechnen.

$$P = \frac{Q}{4\pi R \kappa} \epsilon^- = \frac{D}{\kappa B} M \epsilon^- \quad (4.31)$$

$$\sigma_{ij} = \mu M^* \left[(\mathfrak{I}^- \epsilon^-) \delta_{ij} - \frac{x_i x_j}{R^2} (3\mathfrak{I}^+ \epsilon^-) \right] \quad (4.32)$$

mit

$$R = \sqrt{r^2 + z^2} \quad (4.33)$$

$$\Re = \frac{R}{2\sqrt{Dt}} \quad (4.34)$$

$$M^* = \frac{Q_f \bar{B}}{4\pi R D} \quad (4.35)$$

$$\bar{B} = \frac{B}{3} \frac{1+\nu_u}{1-\nu_u} \quad (4.36)$$

$$\epsilon = \operatorname{erf}(\Re) = \frac{2}{\sqrt{\pi}} \int_0^{\Re} \exp(-s^2) ds \quad (4.37)$$

$$\epsilon^- = 1 - \operatorname{erf}(\Re) = \operatorname{erfc}(\Re) \quad (4.38)$$

$$\Im = \frac{\epsilon / \Re - \dot{\epsilon}}{2\Re} \quad (4.39)$$

Einsetzen der Lösungen für P (4.31) und σ_{ij} (4.32) in die poroelastischen Spannungs-Dehnungs-Beziehung (4.22), (4.23) und Auflösen nach den Deformationskomponenten liefert die Gleichungen:

$$\Delta_V = M^* \epsilon^- = \frac{\kappa \bar{B}}{D} P \quad (4.40)$$

$$\Delta_S = -M^* \frac{3rz}{R^2} \frac{(3\Im + \epsilon^-)}{4} \quad (4.41)$$

$$(\Delta_{r,0} = \Delta_{r,z=0}) \quad \Delta_{r,0} = \frac{M^*}{2} \left[(\Im + \epsilon^-) - \frac{r^2}{R^2} (3\Im + \epsilon^-) \right] \quad (4.42)$$

$$\Delta_T = \frac{M^*}{2} (\Im + \epsilon^-) \quad (4.43)$$

Die Deformation bei konstanter Fluidinjektion stellt sich wie in den zwei folgenden Abbildungen 4.9 und 4.10 ein. In Abb. 4.8 ist die Abhängigkeit der Deformationskomponenten von der Amplitude über dem Abstand der Punktquelle dargestellt. Mit zunehmendem Abstand sinken die Amplituden von Δ_V , Δ_T und Δ_S , die radiale Komponente Δ_r steigt erst mit dem Abstand und sinkt dann wieder.

In Abb. 4.10 sind wiederum die Amplituden der Deformationskomponenten über der Zeit dargestellt. Mit zunehmender Zeit steigen die Amplituden für Δ_V , Δ_T und Δ_S . Die Amplitude der radialen Komponente steigt bis zu dem Zeitpunkt 1 an und fällt daraufhin wieder ab.

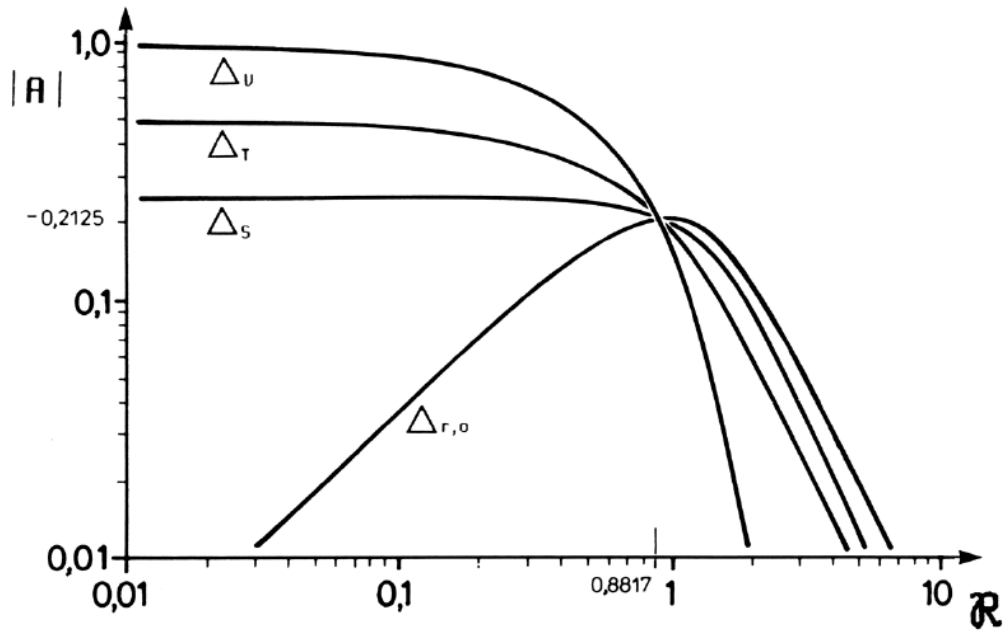


Abb. 4.9: Abhängigkeit der Größen Δ_v bzw. $P, \Delta_{r,0}, \Delta_T$ und Δ_s vom dimensionslosen Abstand $\mathfrak{R} = R / (4Dt)^{1/2}$ in der Umgebung einer Punktquelle konstanter Injektionsrate. Die Amplituden A sind auf $M^* = QB / (4\pi RD)$ normiert (nach Kümpel, 1989).

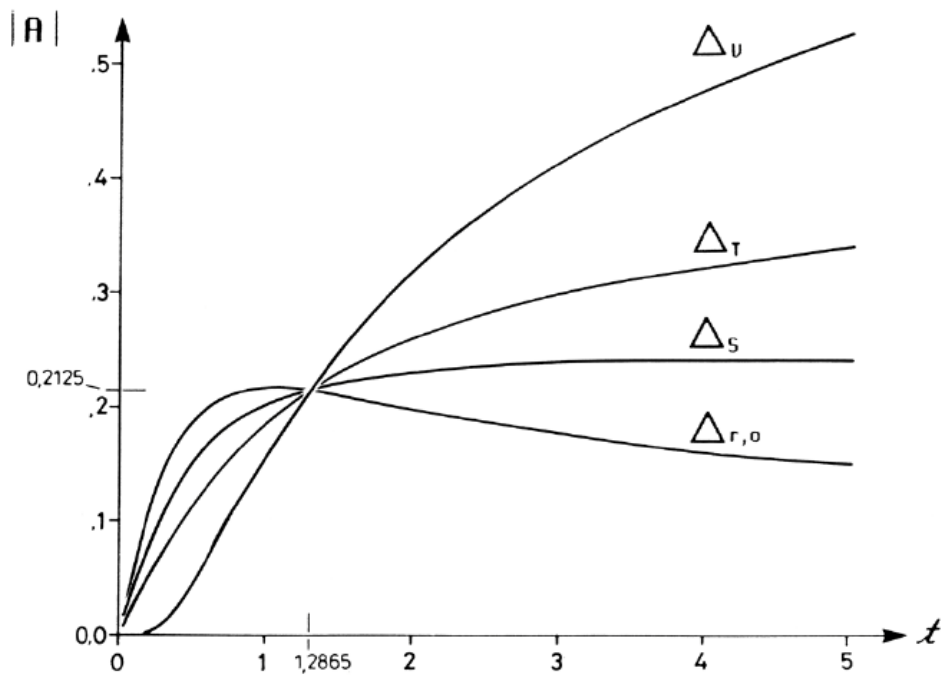


Abb. 4.10: Abhängigkeit der Größen Δ_v bzw. $P, \Delta_{r,0}, \Delta_T$ und Δ_s von der dimensionslosen Zeit $t^* = 4Dt / R^2$ in der Umgebung einer Punktquelle konstanter Injektionsrate. Die Amplituden A sind auf $M^* = QB / (4\pi RD)$ normiert (nach Kümpel, 1989).

5 Messgeräte, Durchführung der Messungen und Sicherung der Daten

5.1 ASKANIA-Bohrlochneigungsmesser

Der Wunsch, Neigungen in der oberen Kruste beobachten zu können, führte zum Bau von Sensoren verschiedener Bauarten (sog. Vertikalpendel, Horizontalpendel, Schlauchwaagen, u.ä., s. Abb. 5.1 a). Vertikalpendel zeichnen sich dadurch aus, dass man sie in Bohrlöchern installieren und dabei störende Effekte der Oberfläche verringern kann. Die Idee Pendel als Messgeräte einzusetzen griff Graf (1958) wieder auf und schlug vor, ein zwei bis vier Meter langes Vertikal-Pendel in einem Rohr in einem Stollen einzubauen. Schneider (1965) baute daraufhin ein 2.62 m langes Pendel in einem Stollen ein und beobachtete gezeitenbedingte Lotschwankungen. Funktionsfähige Geräte wie der ASKANIA-Bohrlochneigungsmesser (Tiltmeter) mit einer Sensorlänge von etwa 60 cm wurden jedoch erst ab 1960 konstruiert (Graf, 1964). Diese Vertikalpendel messen die Neigungsänderung des Bohrlochs gegenüber der Lotrichtung (Abb. 5.1 a).

In Bohrlöchern vor Umwelteinflüssen geschützt, registrieren diese Instrumente Neigungsänderungen mit hoher Genauigkeit und Langzeitstabilität.

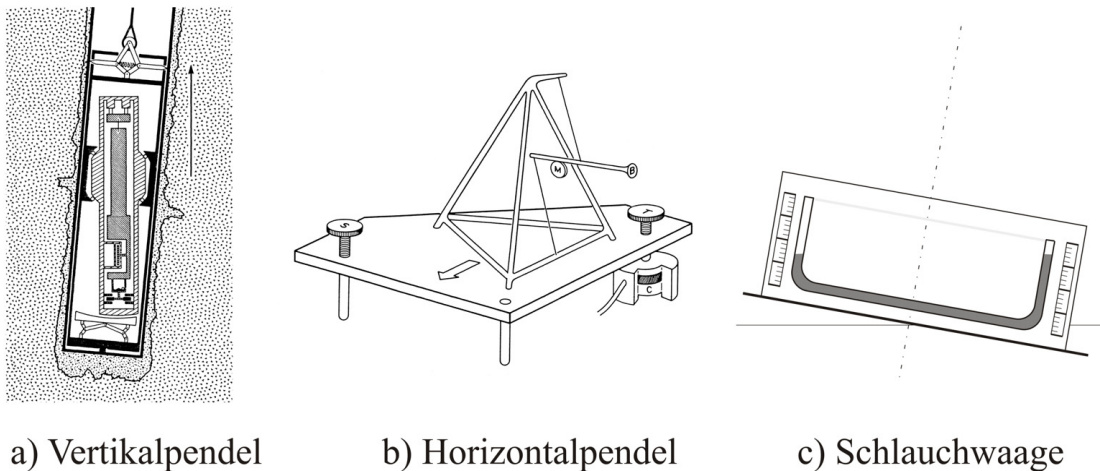


Abb. 5.1: Grundprinzip zur Beobachtung von Neigungen (a und b nach Agnew, 1986; c nach Vogel, 1979).

Sogenannte Horizontalpendel bestehen aus einem horizontalen Arm mit unterschiedlichen Aufhängungen, der senkrecht zur Bildebene schwingt. Schlauchwaagen können sowohl aus einem langen Rohr als auch aus durch Schläuche verbundenen Töpfen bestehen; hierbei kommen neben Wasser unterschiedliche Flüssigkeiten zum Einsatz.

5.1.1 Aufbau

Die hier verwendeten Messgeräte sind Bohrlochneigungsmesser, welche in den 60er Jahren des 20. Jh. von der Firma ASKANIA in Berlin konstruiert und gebaut wurden. Ende der 80er Jahre übernahm das BODENSEEWERK in Überlingen die Produktion. Der ASKANIA-Bohrlochneigungsmesser ist prinzipiell ein nicht astasiertes Vertikalpendel. Diese Art Neigungsmesser besteht grundsätzlich aus zwei Einheiten, der Registrier- und Steuereinheit und dem eigentlichen Messgerät, dem Neigungssensor. Der Neigungssensor wird von einem 1.6 m langen Stahlzylinder geschützt. Dabei wiegt der Neigungsmesser insgesamt 80 kg. Am Kopf des Geräts (Abb. 5.2) befinden sich die Anschlüsse für die X- und Y-Komponente (Abb. 5.2 (1) – X-Komponente; (2) – Y-Komponente). Die angeschlossenen orangefarbene Kabel führen zum Steuerkasten. Die farbigen LED's (Abb. 5.2 (3)) dienen zur optischen Bestimmung der Orientierung des Sensors in der Einbautiefe im Bohrloch (Azimut).

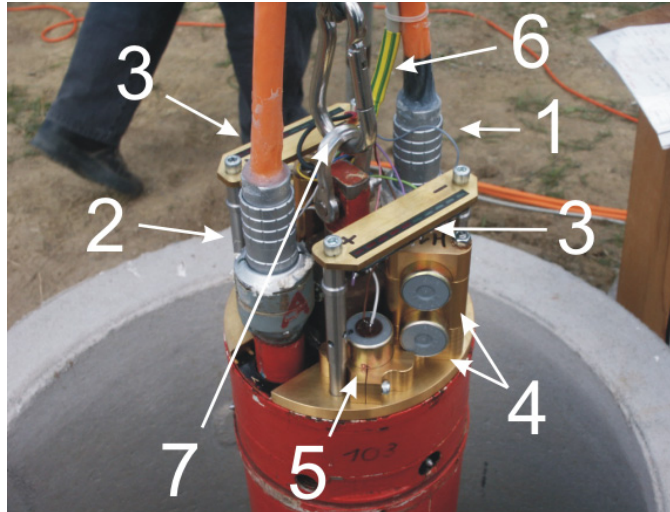


Abb. 5.2: Pendelkopf eines ASKANIA-Bohrlochneigungsmessers mit aufgesetzten Geophonen und LED's. Erklärung der Beschriftung: Siehe Text.

Zur Registrierung wurden jeweils zweimal zwei Horizontalgeophone und zwei Vertikalgeophone installiert (Nr. 4 und 5 in Abb. 5.2) Diese Installation wurde in Rahmen der Arbeiten von Asch et al. (2003) ausgeführt. Das Kabel mit der Beschriftung (6) in Abbildung 5.2 ist die Verbindung zwischen den 4.5 Hz Geophonen und der Registriereinheit, einem „Earth Data Logger“ (EDL).

Der schematische Aufbau eines ASKANIA-Bohrlochneigungsmessers ist in Abb. 5.3 dargestellt. Die unteren Meter der Bohrlochverrohrung bestehen aus einer Edelstahlbuchse (3). Den Abschluss bildet ein Boden (1) auf den ein Zentrierzapfen (2) geschweißt ist. Wird der Neigungsmesser mittels Stahlseil (mit Karabinerhaken Abb. 5.2 (7)) in das Bohrloch gelassen, greift der Zentrierzapfen (2) in die kegelförmige Einbuchtung der Bodenplatte (4) und fixiert den unteren Teil des Instrumentes. Nach dem Aufsetzen wird die vorher durch das Eigengewicht des Gerätes gespannte Spreizfeder (22) entspannt und drückt dabei die drei Spreizspitzen (20) radial nach außen, wodurch die obere Fixierung an der Montageverrohrung (3) erfolgt. Damit liegt die wirksame Basislänge bei 1.4m.

Da ein Bohrloch nie ganz vertikal abgetäuft wird, ist es nötig diese Abweichung vom Lot zu kompensieren. Dazu ist der Pendelträger (Abb. 5.3, 15) mittels einer sphärischen Kugellagerkonstruktion (13, 14) kardanisch am Außenrohr (5) aufgehängt. So können Abweichungen von maximal 3° aus der Vertikalen des Bohrloches automatisch nach Lösen der Arretierung des Pendelträgers (15) ausgeglichen werden. Steht der Pendelträger (15) dann lotrecht, wird der Kreuzschlitten (8) durch den motorgetriebenen

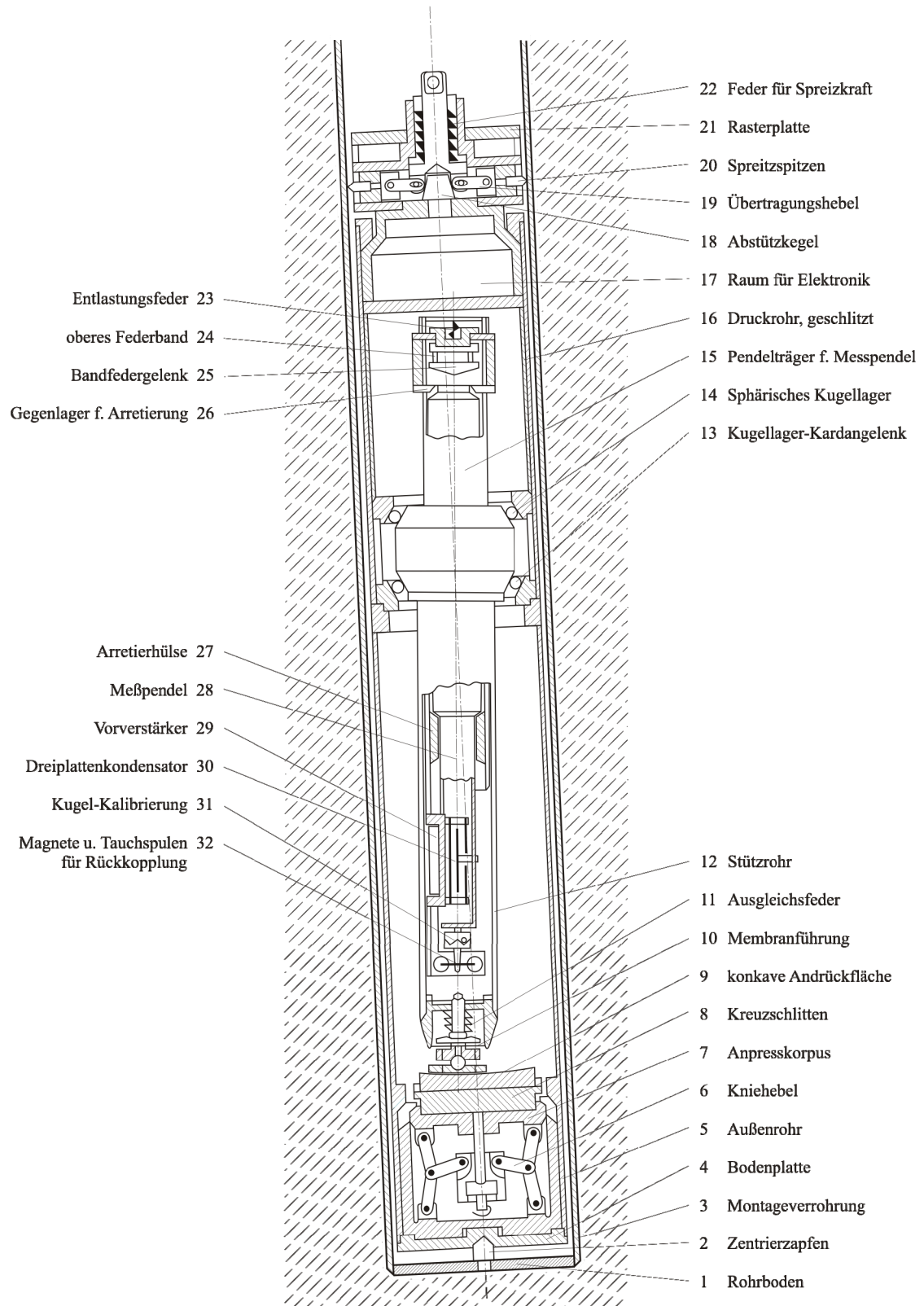


Abb. 5.3: Aufbau des ASKANIA-Bohrlochneigungsmessers (Jacoby, 1966)

Anpressmechanismus (6, 7) zur Arretierung an die Unterseite des Pendelträgers (15) gedrückt. Zur Feinjustierung des Pendelträgers im Bereich von 13 Bogenminuten kann der Kreuzschlitten (8) in zwei orthogonale Richtungen bewegt werden. Der Pendelträger (15) und der Kreuzschlitten (8) sind starr verbunden. Anschließend wird das Messpendel (28) aus seiner Arretiervorrichtung (26, 27) gelöst. Das Messpendel (28) ist über ein Kreuzfedergelenk (25), von zwei senkrecht zueinander stehenden Metallplättchen am Pendelträger (15) befestigt. Damit hat das Gelenk einen Drehpunkt aber zwei Drehachsen, wodurch laut Hersteller das Rückstellmoment dieser Konstruktion 10^3 -mal kleiner als das auslenkende Drehmoment ist (ASKANIA, 1968).

Am unteren Ende des Messpendels befinden sich zwei Tauchspulen (32), zwei Kugel-Kalibrier-Einrichtungen (31) für die beiden Komponenten und zwei Dreiplattenkondensatoren (30). Die Tauchspulen wurden ursprünglich zur elektrischen Kalibrierung verwendet und dienen seit dem Umbau durch Zschau (1974) der elektromagnetischen Rückkopplung zur Dämpfung von Signalen nahe der Sensor-Eigenperiode von 1.4 s. Die Kugel-Kalibrierung und ihre Funktionsweise werden in Kapitel 5.1.2 ausführlich beschrieben.

Die Umwandlung des Ausschlags des Neigungsmessers in einen Messwert erfolgt mit Hilfe zweier zueinander orthogonaler Dreiplattenkondensatoren. Von diesen jeweils drei Platten ist die mittlere starr mit dem Sensor (Messpendel) verbunden. Die beiden Äußeren sind am Pendelträger befestigt. Jede Bewegung der Mittelplatte dieser kapazitiven Brücke verändert die Lage der Platten zueinander und beeinflusst damit die Kapazitäten. Eine ausführliche Beschreibung des kapazitiven Abgriffsystems ist bei Asch (1988) zu finden. Wichtig ist, dass eine Bewegung senkrecht zur Messrichtung keine Kapazitätsänderung bewirkt, so dass also zwei zueinander senkrechte Richtungen an einem Sensor abgegriffen werden können. Die Empfindlichkeit dieser Schaltung liegt laut Hersteller bei $10 \mu\text{m/V}$ für Plattendistanzen von 0.5 mm. Diese kapazitive Brücke wird über ein Trägerfrequenzverfahren mit 10 kHz in eine Gleichspannung umgewandelt. Diese wird an den Vorverstärkern (29) vergrößert und dann über die Messkabel an den Schaltkasten weiter geleitet.

5.1.2 Ankopplung

Zur Gewährleistung der Ausnutzung der hohen Auflösung wie auch der Langzeitstabilität des ASKANIA-Bohrlochneigungsmessers ist es notwendig, dass das Gerät optimal installiert wird und die Anbindung an das Umgebungsgestein spannungsfrei ist (Weise, 1992).

Nahe der Oberfläche stören meteorologisch induzierte Effekte, vor allem Schwankungen des Grundwassers, die Neigungsregistrierungen empfindlich. Deshalb wird der ASKANIA-Neigungsmesser im Bohrloch betrieben, wobei Installationstiefen von 30 m oder mehr empfohlen werden (Jentzsch, 1993). Damit werden zugleich auch konstante Bedingungen in der Temperatur ermöglicht. Die Installationstiefen der Neigungsmesser an den Stationen sind der Tabelle 5.1 zu entnehmen.

Tab. 5.1: Installationstiefen der Neigungsmesser der Stationen.

Station	Unterkante der Neigungsmesser in m	Tiefe Grundwassermessstelle in m
Berg	30.5	30.5
Eiglasdorf	44.0	51.0
Stockau	40.6	42.0
Mittelberg	45.5	45.0
Püllersreuth	24.5	42.3

Neigungssignale durch Niederschlag und Grundwasser machen sich in dieser Tiefe jedoch immer noch, wenn auch geringer, bemerkbar (Jentzsch, 1993). Deshalb wurden zusätzlich an jeder Station Grundwassermessstellen und an der KTB eine meteorologische Station eingerichtet, um diese Parameter mit zu beobachten und die Einflüsse bestimmen zu können.

Strain-Tilt-Kopplungen können Neigungsamplituden um ein Vielfaches gegenüber den theoretischen Gezeiten erhöhen und zu Phasenverschiebungen von einigen Grad führen (Harrison, 1976). Diese Kopplungen können durch eine entsprechende Standortwahl minimiert werden. Das Material in dem der Neigungsmesser installiert wird, sollte isotrop und elastisch sein und das Bohrloch (Kapitel 2) gegenüber der freien Oberfläche rechtwinklig. Leider können diese Bedingungen in der Praxis nie ganz erfüllt werden.

5.1.3 Kalibriervorrichtung

Die Kalibrierung dient der Umrechnung der beobachteten Neigung von Millivolt (mV) in Millibogensekunden (msec) oder Radian (rad). Bei diesen Geräten umfasst die Kalibrierung den gesamten Signalweg vom Sensor bis hin zur Registrierung. Durch die wiederholte Kalibrierung lassen sich Langzeitstabilität und Linearität überprüfen (Kapitel 5.1.5). Der größte Vorteil ist, dass die in situ erfolgende Kalibrierung den normalen Messvorgang nicht unterbricht und somit keine Lücken in den Daten entstehen.

Eine genau bestimmte Masse (Stahlkugel) wird definiert in ihrer Lage innerhalb eines Kugelkäfigs verändert, womit eine Auslenkung des Pendels hervorgerufen wird. Für beide Kanäle sind eigene Kalibriervorrichtungen vorhanden, die orthogonal zueinander gelagert sind. In einer mit dem Messpendel fest verbundenen Kammer befindet sich eine Stahlkugel mit einer Masse von ca. 32 mg (Abb. 5.4), die genau zwei definierte Lagen einnehmen kann. Die Lageänderung wird über den Schaltkasten ausgelöst und mittels der am Kugelkäfig angebrachten Elektromagneten vorgenommen. Durch den elektromagnetischen Impuls eines Elektromagneten wird die Kugel in Mittellage gezogen und gleitet durch ihre Trägheit in die andere Position (ASKANIA, 1968).

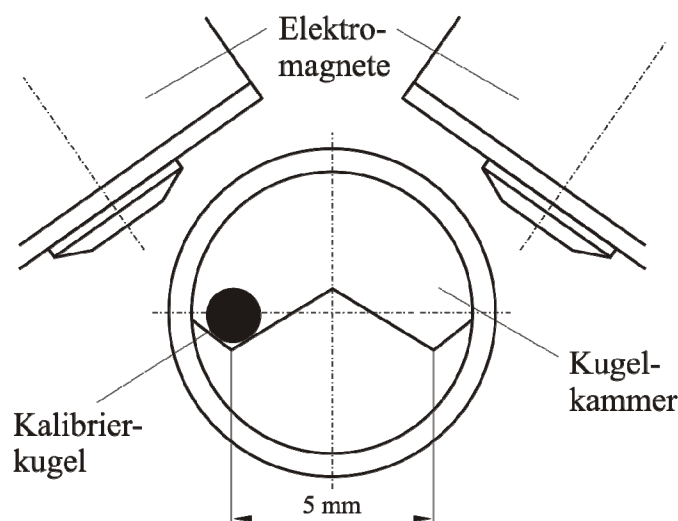


Abb. 5.4: Aufbau der Kalibriervorrichtung (nach Jacoby, 1966).

Über die Kugelmasse m , dem Abstand der beiden Ruhelagen A und dem Produkt P aus Kugelmasse und Schwerpunktabstand zur Aufhängung kann die absolute Winkeländerung φ bei einem Kalibriersprung mit

$$\varphi = \frac{(m \cdot A)}{P} \quad (5.1)$$

berechnet werden. Ein Kalibriersprung beträgt circa 30 ms entsprechend etwa 300 mV in den registrierten Daten (Abb. 6.13). Damit entspricht die Sprungamplitude in etwa der Größenordnung der Gezeitenneigungen in mittleren Breiten.

Die notwendigen genauen Werte für Kugelmasse, Abstand der Ruhelagen und Schwerpunktabstand wurden für jeden ASKANIA-Bohrlochneigungsmesser einzeln vermessen und sind für die verwendeten Neigungsmesser in Tabelle 5.2 aufgeführt.

Tab. 5.2 : Auszug aus den Messblättern der Neigungsmesser zur Kalibrierung der Datenreihen (Herstellerangaben).

Station	Berg	Eiglasdorf	Stockau		Mittelberg	Püllersreuth
Instrument ID	103	111	108	104	112	102
Kugelkalibrierung X-Kanal						
Winkelausschlag (msec/mV)	50.01	33.70	31.70	30.78	49.80	20.63
Kugelkalibrierung Y-Kanal						
Winkelausschlag (msec/mV)	49.41	33.30	31.91	46.55	32.10	20.63

Aus dem bekannten absoluten Winkelausschlag φ in Millisekunden und der beim Kalibriersprung registrierten Spannungsänderung U in Millivolt lässt sich der Kalibrierfaktor K als Quotient von φ und U ermitteln.

$$K = \frac{\varphi}{U} \quad (5.2)$$

Durch die Kalibrierung lässt sich die Stabilität der Messungen insbesondere bei Langzeitregistrierungen überprüfen und sollte deshalb wöchentlich vorgenommen werden. Notwendig ist eine Neubestimmung der Kalibrierfaktoren bei jeder Neuinstallation von Komponenten der Messapparatur. Laut Herstellerangaben lässt sich mit dieser Kalibrierung eine theoretische Genauigkeit von maximal $\pm 0.2\%$ erzielen. Diese Genauigkeit hängt dabei von der Anzahl der Kalibriersprünge pro Zeiteinheit und der digitalen Datenerfassung ab. Die Genauigkeitsgrenze wird durch die Amplitude der Mikroseismik und die Abtastrate festgesetzt (Große-Brauckmann, 1979; Weise, 1992).

Einige Anwender haben ein Übersprechen des Kalibriersprunges in einem Kanal auf die Registrierung des anderen Kanals von bis zu mehreren Prozent festgestellt (Johnson et al., 1995; Gomez et al., 1995; Weise, 1992) (Abb. 6.13). Mögliche Ursachen dafür sind: nicht orthogonale Stellung der Abgriffsysteme, nicht genaue Entsprechung der Neigungsrichtung der Orientierung des Kugelkäfigs und nicht parallele Be-

wegung der Kugel im Käfig (Weise, 1992). Die Höhe des Übersprechens war bei den einzelnen Instrumenten unterschiedlich, jedoch so groß, dass eine Korrektur wie unter Kapitel 6.2.1.2 beschrieben erforderlich war.

5.1.4 Azimutbestimmung

Die ASKANIA-Bohrlochneigungsmesser messen Neigungen in zwei orthogonalen Komponenten. Deren Orientierungen fallen jedoch fast nie mit den für alle Betrachtungen verwendeten Himmelsrichtungen Nordsüd und Ostwest zusammen (bzw. Nordost-Südwest und Nordwest-Südost, welche in manchen Fällen betrachtet werden). Die im System von X und Y gemessenen Neigungen müssen auf ein Koordinatensystem der Erdoberfläche transformiert werden (Kapitel 6.2.4), wozu bereits bei der Installation der Azimutwinkel bestimmt wird. Die Bestimmung der Azimutwinkel der Neigungsmesser erfolgte optisch mit einem Nivelier Ni-2, auf das ein 90° Prisma aufgesteckt ist. Auf dem Kopf der Bohrlochneigungsmesser (Abb. 5.5) befinden sich neben den Anschlüssen für die Kanäle und der Aufhängevorrichtung eine Markierung und zwei parallele Leuchtdioden-Ketten. Die Markierung „0“ bezeichnet die Messrichtung des Y-Kanals. Die zwei zur Hälfte grünen und roten Leuchtdioden verlaufen parallel zur X-Messrichtung. Zwischen der Markierung „0“ und dem Y-Kanal wurden für einige Neigungsmesser Abweichungen von einigen Grad festgestellt (Gomez et al., 1995). Nach der Installation des Neigungsmessers werden die Leuchtdioden mit Hilfe des optischen Messgeräts direkt auf der Öffnung des Bohrlochs auf ein terrestrisches oder astronomisches Ziel eingemessen. Die erreichbare Genauigkeit von $\pm 0.1^\circ$ bis $\pm 1^\circ$ hängt von der Tiefe des Bohrlochs und den Sichtverhältnissen ab (Weise, 1992), wobei der Winkel von Nord aus rechtsdrehend gemessen wird.

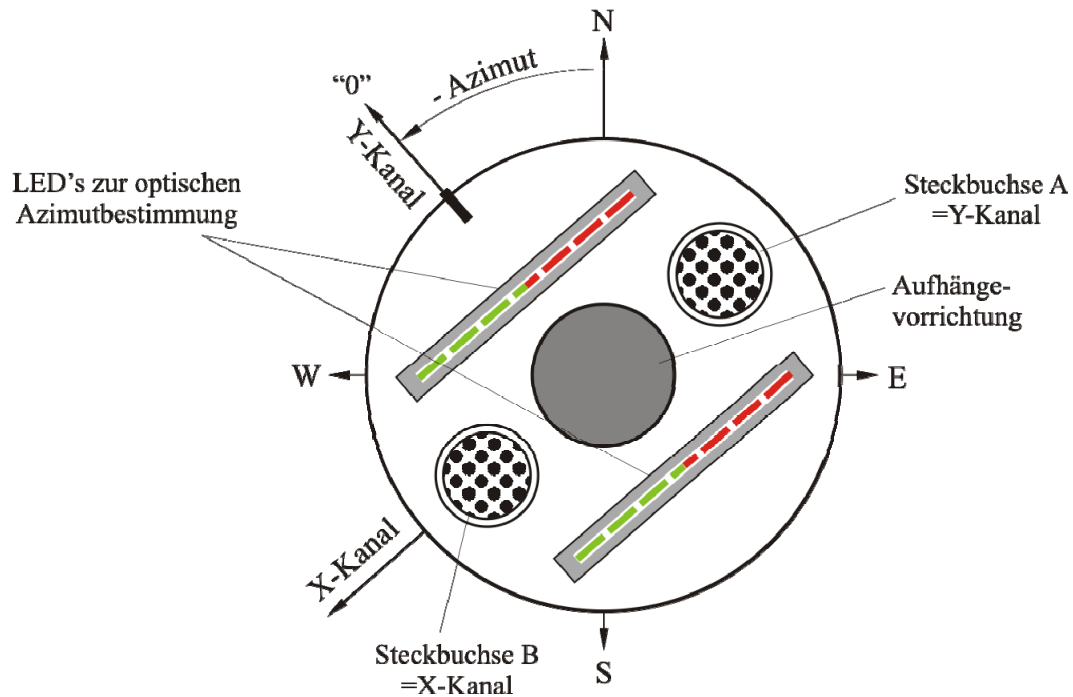


Abb. 5.5: Aufbau der Deckplatte des ASKANIA-Bohrlochneigungsmessers zur Azimutbestimmung.

Wie in Abb. 5.2 zu sehen ist, sind auf dem Pendelkopf drei Geophonsätze (4, 5) zur Registrierung seismischer Signale installiert (Asch et al., 2003), die ebenfalls zur Azimutbestimmung verwendet werden können. Bestimmt wurden die seismischen Azimute der Stationen von Jörn Kummero (GFZ). In Tabelle (5.3) sind die ermittelten Azimute der Stationen zusammengestellt.

Tab. 5.3: Azimute der Neigungsmesserstationen an der KTB.

Station	Gültiger Zeitraum		Instrument ID	Transformationswinkel des Y-Kanals in Richtung Nord in °
	von	Bis		
Berg	02.09.'03	-	103	135.78
Eiglasdorf	15.10.'03	-	111	138.00
Stockau	01.08.'03	31.08.'04	108	172.00
Stockau	31.08.'04	-	104	331.00
Mittelberg	02.09.'03	-	112	310.00
Püllersreuth	21.07.'03	-	102	345.00

Bei der Station Stockau war es im August 2004 wegen eines Blitz-Schadens notwendig geworden, den Neigungsmessers zu wechseln. Deshalb musste auch das Azimut des Pendels neu bestimmt werden.

5.1.5 Auflösung, Linearität, Langzeitstabilität der Registrierungen

Die Auflösung der Neigungsmesser beschreibt die kleinste, zuverlässig mess- und reproduzierbare Auslenkung der Pendelspitze. Nach Große-Brauckmann & Rosenbach (1979) ist die Auflösung für Neigungsänderungen im günstigsten Fall 0.01 ms. Unter dem günstigsten Fall versteht man zum Beispiel die unter Kapitel 5.1.4 beschriebene Ankopplung der Messgeräte, aber auch das Registriersystem mit seiner Dynamik. Das Bauelement mit der geringsten Dynamik ist der limitierende Faktor für das real aufgezeichnete Signal. In Abhängigkeit von Rauschpegel und der Reproduzierbarkeit der Messwerte sollte das Registriersystem gewählt sein. Mehr dazu findet sich unter Kapitel 5.1.7. Die ideale Auflösung des Neigungsmessers von 0.01 ms entspricht bei einer Pendellänge von 60 cm einer Auslenkung der Pendelspitze um 0.03 nm oder anschaulicher, der Auslenkung eines vom Mond zur Erde reichenden Pendels um 4 m. Die Bewegung der Pendelspitze kann also auf eine Entfernung aufgelöst werden, die weniger als dem Abstand der Atome im NaCl-Gitter entspricht.

Um die maximale Auflösung sinnvoll nutzen zu können, ist es notwendig, sich im linearen Messbereich zu bewegen. Der gesamte Messbereich der Neigungsmesser umfasst ± 10 V. Der als linear zu betrachtende Bereich des Messgerätes umfasst aber nur ± 3 V um die Nulllage (ASKANIA, 1968). Für Langzeitbeobachtungen ist eine hohe Linearität notwendig. Nach Große-Brauckmann & Rosenbach (1979) ist die Linearität der Neigungen im Bereich von ± 150 msec besser als ± 0.1 %.

Die Langzeitstabilität wird von zwei Faktoren beeinflusst. Der eine Faktor ist die Stabilität des Signalabgriffs und die folgende Signalverstärkung, der zweite ist die Stabilität des Messgerätes selbst. Der erste Faktor lässt sich leicht durch die Kalibrierung überprüfen. Der Zweite ist weit schwieriger abzuschätzen, da sich instrumentelle Drift und geodynamisches Signal überlagern. Die Drift sollte jedoch unter 50ms pro Jahr liegen (Große-Brauckmann & Rosenbach, 1979).

5.1.6 Übertragungseigenschaften

Die Neigungsmesser haben bestimmte Übertragungseigenschaften. Diese wurden von Große-Brauckmann (1979) und Asch (1988) ausführlich untersucht, welche auch die dazugehörige Charakteristik ermittelten.

Neigungsmesser reagieren sowohl auf Neigungen als auch auf horizontale Beschleunigungen (Kapitel 4.1). Die Eigenfrequenz der ASKANIA-Bohrlochneigungsmesser liegt bei 0.7 Hz, was einer Periode von 1.4 s entspricht. Aus dem Aufbau der Geräte variieren diese Werte leicht für beide Komponenten. Bedingt durch die geringe Dämpfung der Neigungsmesser von nur 0.01 wird das Instrument im Bereich der Eigenfrequenz durch Mikroseismik zur Resonanz angeregt. Die Amplitude steigt dabei um den Faktor 50. Die Resonanz kann durch Zuschalten einer elektromagnetischen Rückkopplung (Zschau, 1974) gedämpft werden, diese ist aber an den Stationen nicht aktiviert. Für den Einsatz an der KTB werden hauptsächlich die langperiodischen Anteile (Gezeiten, langperiodische Signale / Driften) verwendet. Dort zeigt laut Große-Brauckmann & Rosenbach (1979) der Neigungsmesser eine flache Frequenzantwort. Zwar sind die Beschleunigungen und Neigungen nicht voneinander trennbar, jedoch bei Perioden oberhalb von 1.4 s wirkt das Instrument in erster Linie als Neigungsmesser.

5.1.7 Registrierung der Daten

Zur Registrierung der Daten ist ein komplexes System von Geräten und Bauelementen nötig, das in Abb. 5.6 dargestellt ist. Zur besseren Übersicht sind auch alle Wege, die die Rohdaten nehmen, verzeichnet. Das Funknetzwerk wurde unter Kapitel 2.3 vorgestellt.

Die gesamte Steuer- und Registriereinheit ist gepuffert, damit kurze Stromausfälle von bis zu 2 Tagen überbrückt werden können. Dazu dienen zwei 12 V Batterien mit 80 Ah Kapazität, die von einem Ladegerät gespeist werden. Die Kontrolleinheit für den Neigungsmesser benötigt eine Spannung von 24 V. Dazu dient ein Gleichstromwandler, der die Spannung von 12 V auf 24 V transformiert.

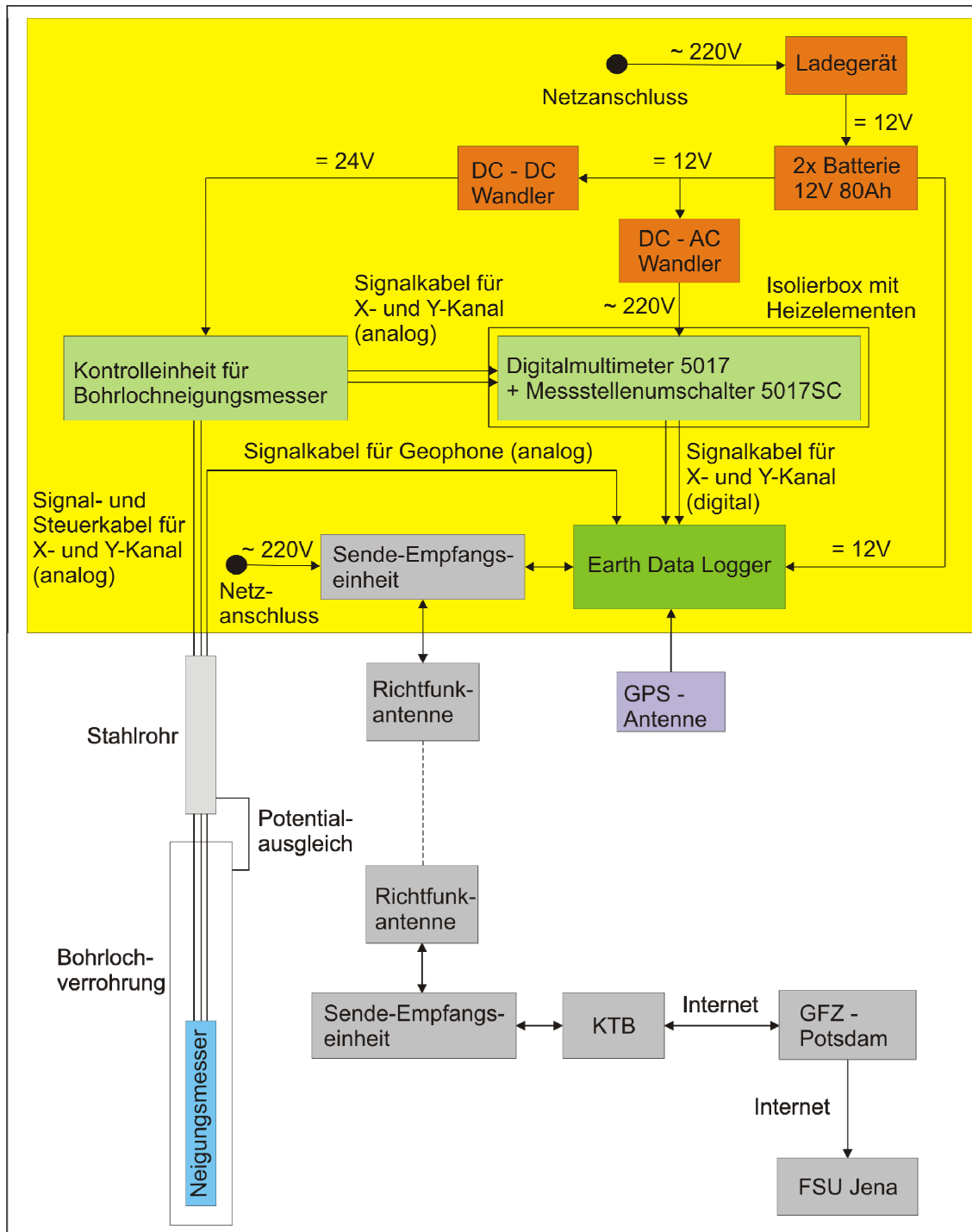


Abb. 5.6: Schema der Messapparatur und der Datenwege.

Über die Kontrolleinheit wird zum einen der Neigungsmesser gesteuert und zum anderen werden die Signale der Kanäle abgegriffen. Die Kontrolleinheit enthält zudem wahlweise einen 8- oder 80-Sekunden Tiefpassfilter, welcher das Eingangssignal der Kanäle analog filtert. Im vorliegenden Fall wird der 8-Sekunden-Filter verwendet. Der Grund dafür ist, dass die notwendigen Schaltkreise für den 80-Sekunden Filter sehr alt

sind und nicht gewährleistet ist, dass diese optimal arbeiten. Nach Ramatschi (1994) liegt die Verzögerung des Eingangssignals im Filter bei 13 s für den 8-Sekunden-Filter, wodurch er bei einer Abtastrate von 1 Wert pro 10 Sekunden als Aliasfilter akzeptabel ist, da die zu erwartenden Signale in diesem Periodenbereich sehr klein und nicht rein periodisch sind. Der 80-Sekunden-Filter kann bereits zu merkbaren Phasenverschiebungen bei den Gezeiten führen.

Nach der analogen Filterung werden die Signale mit Hilfe des A/D-Wandlers (Digitalmultimeter 5017 von der Firma PREMA Semiconductor) digitalisiert. Da das Digitalmultimeter immer nur einen Kanal verarbeiten kann, wird intern über den Messstellenumschalter alle fünf Sekunden der Kanal gewechselt, womit man eine Abtastrate der Daten von 10 s erhält. Der A/D-Wandler benötigt einen Netzanschluss. Dafür werden die 12 V aus dem Puffersystem auf 220 V Wechselspannung transformiert.

Der Wandler digitalisiert mit 24 bit (Kapitel 5.1.5). Die nun digitalen Signale werden durch den „Earth Data Logger“ (EDL) nur aufgezeichnet, ohne dessen A/D-Wandlung zu nutzen. Dazu wird für die Zeitkennung über eine GPS-Uhr der EDL das Zeitsignal (UTC) eingespeist. Die EDL erfasst (und digitalisiert) zudem die Signale der drei Geophone, die auf dem Kopf des Neigungsmessers (Kapitel 5.1.1) angebracht sind. Die Daten werden über das Funknetz zur KTB gesendet und dort zunächst gespeichert. Von dort gelangen sie dann via Internet an das GFZ-Potsdam und an die FSU-Jena. Das GFZ hat zudem bei laufendem Funknetz vollen Zugriff auf die Daten via Internet.

5.2 Diver – Grundwasser-Messsensoren

Zur Berücksichtigung der in Kapitel 5.1.4 erwähnten meteorologischen Effekte wurden an jeder Station Grundwassermessstellen eingerichtet. Die Teufen der Bohrlöcher sind in Tabelle 5.1 zusammengefasst. Die eingebauten Sensoren sind sog. Diver der Firma Van Essen, die über ein eigenes Datenerfassungssystem verfügen. Tabelle 5.4 zeigt die Spezifikationen der verwendeten Geräte.

Tab. 5.4: Spezifikationen der Diver (Herstellerangaben).

Diver® Technische Spezifikationen			
Allgemein:			
Garantie	Standard 3 Jahre		
Messintervall*	0.5 Sek. bis 99 Stunden		
Speicherplatz	24000 Messungen		
Gehäusematerial	Edelstahl 316L		
Druckgebermaterial	Keramik		
Temperaturbereich	-20°C bis 80°C		
- Genauigkeit	±0.1°C		
- Auflösungsvermögen	0.01°C		
- Kompensierter Bereich	-10°C bis 40°C		
Batterielebensdauer	8-10 Jahre		
Abmessung	Ø22 mm, Länge 125 mm		
Gewicht	160 g		
Typ	DI 240	DI 241	DI 242
Kalibrierter Bereich	5 m Wassersäule	10 m Wassersäule	20 m Wassersäule
Nutzbereich	4 m Wassersäule	9 m Wassersäule	19 m Wassersäule
- Genauigkeit**	±0.1% ME	±0.1% ME	±0.1% ME
- Auflösungsvermögen	0.1 cm	0.2 cm	0.4 cm
Typ	DI 243	DI 245	DI 250 (Baro)
Kalibrierter Bereich	30 m Wassersäule	100 m Wassersäule	1,5 m Wassersäule
Nutzbereich	29 m Wassersäule	99 m Wassersäule	--
- Genauigkeit**	±0.1% ME	±0.1% ME	±0.3% ME
- Auflösungsvermögen	0.6 cm	2 cm	0.1 cm
* mehrere Messmethoden verfügbar (fest, ergebnisabhängig und Pumpversuche)			
** innerhalb des temperaturkompensierten Bereiches			

Die verwendeten Sensoren sind die Diver „DI 250 (Baro)“ und „DI 240“. Das Messgerät ist ca. 20 cm lang (Abb. 5.7). Über das angeschlossene Kabel und einen Laptop bzw. spezielles Handauslesegerät lassen sich die Daten vom Messgerät herunterladen.



Abb. 5.7: Diver mit Auslekabel.

Die sehr kompakten Geräte messen Druckänderung und Temperatur. Der Messbereich beträgt 5 m Wassersäule, wobei die Auflösung bei 0.1 cm liegt. Die Auflösung für die Messung der Temperatur beträgt 0.01°C. Bei der verwendeten Abtastrate von 5 Minuten hat der Sensor eine Registrierdauer von 83.3 Tagen. Die Daten werden aber in einem Intervall von zwei bis vier Wochen heruntergeladen, um, falls der Wasserstand aus dem Messbereich herausgelaufen ist, den Sensor neu zu justieren.



Abb. 5.8: Packer der Firma Comdrill (links - unaufgeblasen, rechts - aufgeblasen).

Durch die Abpackerung (Abb. 5.8) wird nicht die Änderung des Grundwasserspiegels sondern direkt die Porendruckänderung gemessen, ohne dass eine Wasserbewegung im Bohrloch stattfinden muss, wodurch die Registrierungen ein deutlich höherfrequentes Signal zeigen. Die Schlauchpacker stammen von der Firma Comdrill. Die Gesamtlänge ist ca. 1 m, wobei der Gummischlauch ca. 40 cm lang ist. Der Durchmesser des Gummischlauches im aufgeblasenen Zustand beträgt etwa 58 mm.

Der Packer hat im Inneren eine Durchführung für das Datenkabel der Diver (Abb. 5.9 (1)). Um den Packer aufzublasen ist ein Druckschlauch (Abb. 5.9 (2)) und zur Sicherung und leichteren Justierung in der Verrohrung ein Stahlseil angebracht (Abb. 5.9 (3)).

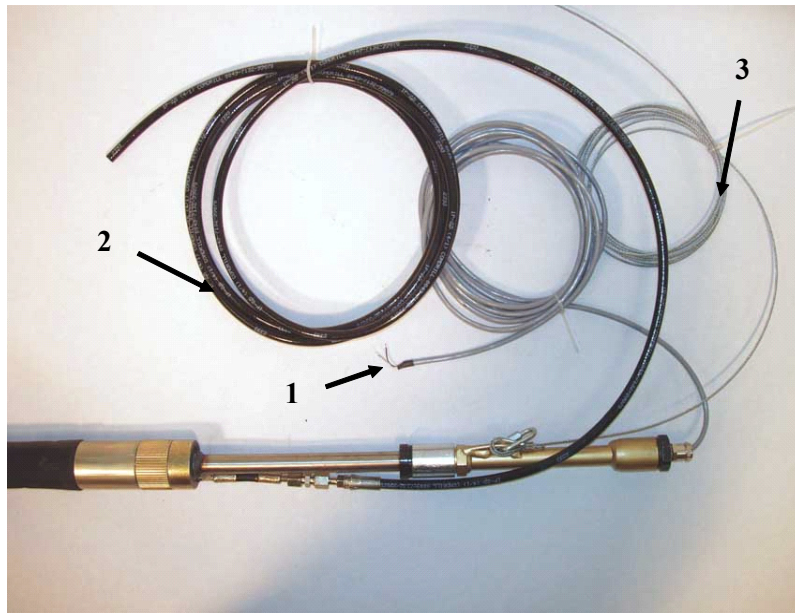


Abb. 5.9: Packer mit Kabeln und Anschlüssen (1 – Datenkabel, 2 – Druckschlauch, 3 – Stahlseil).

Die Packer wurden in den Grundwassermessstellen knapp unterhalb des mit dem Lichtlot gemessenen Pegels gesetzt und mit einem Druck von 10 bar aufgeblasen. Die Diver sind 2 m unterhalb der Packer positioniert.

In Abb. 5.10 ist die Grundwassermessstelle von Mittelberg neben der Instrumentenverrohrung zu sehen, die bereits erwähnten Kabel und Anschlüsse sind hier wieder zu finden. Das Datenkabel ist über einen Adapter (1) mit einem Laptop verbunden, um die gemessenen Daten abzuzeihen. Der Druckschlauch (2) ist mit einem Manometer versehen, um den Druck des Packers zu überprüfen. An der Sicherungskappe (3), ist das schon erwähnte Stahlseil zur Sicherung der Installation angeschlossen. Die Daten werden mit dem von Van Essen gelieferten Programm LDM heruntergeladen und bearbeitet. Um aus den gemessenen Porendruckschwankungen die Pegelstände zu ermitteln, müssen sie mit Hilfe des Luftdrucks kompensiert werden. Dazu werden die Messwerte des Baro-Divers „DI 250“ verwendet. Dies erfolgt wiederum mit Hilfe von LDM.

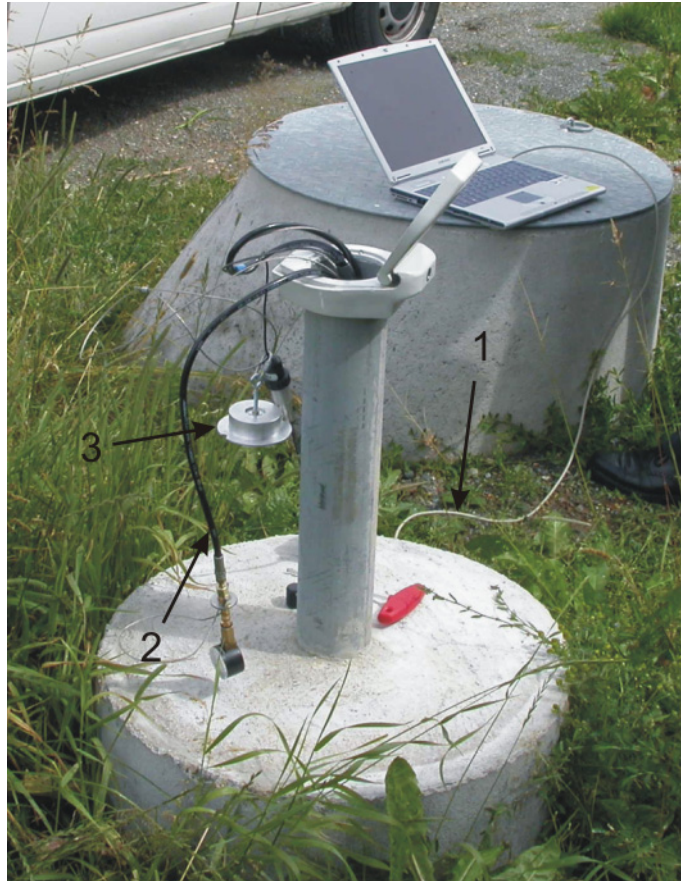


Abb. 5.10: Grundwassermessstelle und Neigungsmesserbohrung Mittelberg.

5.3 Meteorologische Station

Das Wetter hat einen Einfluss auf das beobachtete Neigungssignal. Um diesen Anteil zu separieren, wurde an der KTB eine Wetterstation aufgebaut (Abb. 5.11). Gemessen werden unter anderem die Parameter Temperatur, Niederschlag, Luftdruck, Windrichtung und Windgeschwindigkeit. Bei der Wetterstation handelt es sich um eine Kompaktwetterstation des Herstellers Vantage, USA.



Abb. 5.11: Kompaktwetterstation an der KTB.

Die technischen Spezifikationen der einzelnen Elemente sind in Tabelle 5.5 zusammengefasst. Die Einstellung wurde so gewählt, dass die Auflösung des Niederschlagsmessers mit 0.2 mm hoch ist, da dieser Parameter am meisten interessiert.

Tab. 5.5: Technische Spezifikation der Komponenten der Wetterstation.

Komponente	Messbereich	Auflösung
Windrichtung	0° - 360°	< 7°
Windgeschwindigkeit	0 m/s - 50 m/s	< 0.5 m/s
Lufttemperatur	-40°C - +60°C	0.5 K
Luftfeuchtigkeit	10% - 100%	3%
Luftdruck	880 hPa - 1080 hPa	1 hPa
Niederschlag		0.2 mm mit ±5%

Die Wetterdaten wurden am Anfang im Stundenintervall und später halbstündlich mit einem Datenlogger aufgezeichnet. Diese Daten werden dann wie die Grundwasserdaten mit Hilfe eines Laptops abgezogen. Dies geschieht mit dem Programm *Weatherlink*, wobei mit Hilfe einer GPS-PC Uhr das Zeitsignal auf UTC synchronisiert wird. In dem Programm können die Messwerte graphisch dargestellt oder als Textdatei herausgeschrieben werden. Die weitere Bearbeitung der Daten ist unter Kapitel 6.4 zu finden.

6 Datenauswertung

6.1 Allgemeiner Überblick über die Auswertung von Tiltmeter-Daten

Die Auswertung der Tiltmeter-Daten erfolgt in unterschiedlichen Schritten; in Abb. 6.1 ist der Prozess schematisch dargestellt.

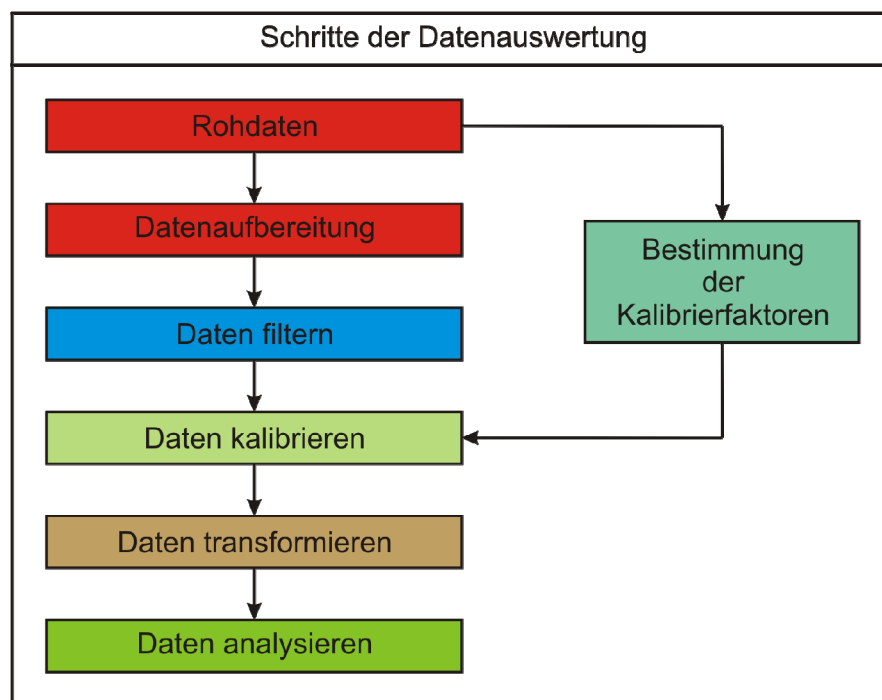


Abb. 6.1: Schematische Darstellung der Auswertung der Tiltmeter-Daten.

Nachdem die Rohdaten gesichtet sind, unterteilt sich die Datenauswertung in fünf Hauptphasen: Die erste ist die Datenaufbereitung, hier werden die Rohdaten zu einer langen kompletten Zeitreihe zusammengefügt. Darauf folgt die zweite Phase, das

Filtern der Datenreihen von 10 Sekunden- auf Stundendaten. Die nächste Phase bezeichnet das Kalibrieren der Datenreihen mit den zuvor bestimmten Kalibrierfaktoren (Kapitel 6.2.3). Als vorletzter Schritt erfolgt die Transformation der Kanäle in Nordsüd-Ostwest-Richtung. Am Ende erfolgt die Analyse zur Ermittlung der Gezeitenparameter.

6.2 Auswertung von Tiltmeter-Daten

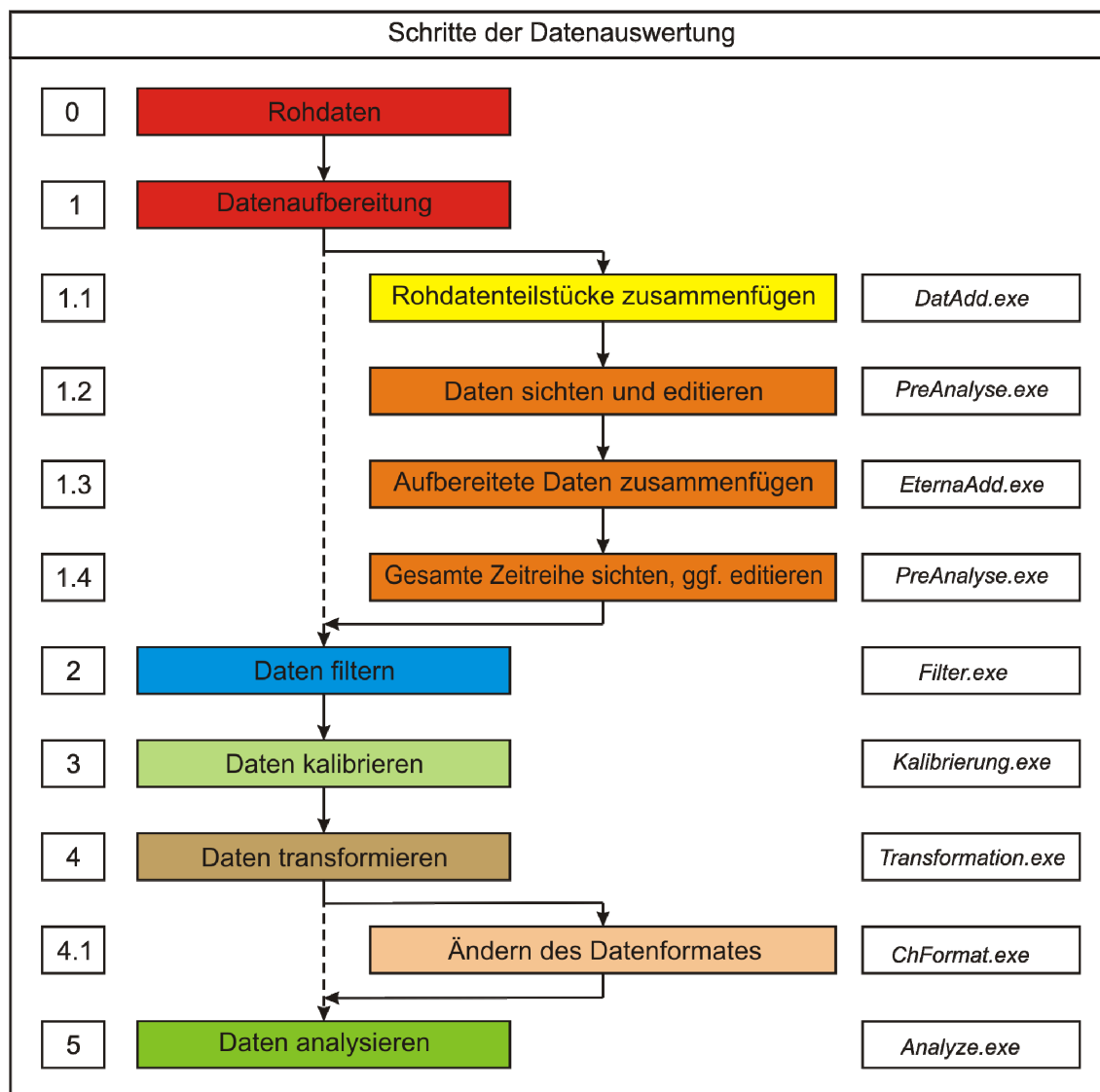


Abb. 6.2: Schematische Darstellung des Prozessing der Tiltmeter-Daten und Benennung der entsprechenden Routinen.

In Kapitel 6.1 wurde der allgemeine Weg der Auswertung von Tiltmeter-Daten beschrieben. In diesem Abschnitt werden die Schritte detailliert erläutert und die verwendeten Programme beschrieben. Zudem sollen einige Probleme bei der Entwicklung und deren Lösungen aufgezeigt werden. Hinter den Einzelschritten sind die verwendeten Programm-Namen aufgelistet.

6.2.1 Datenaufbreitung

Die Datenaufbereitung unterteilt sich in vier Schritte. Nachdem die Rohdaten via Internet vom GFZ gesendet wurden, müssen diese als erstes in ein anderes Datenformat umgewandelt und die einzelnen Tagesdateien zusammengefügt werden. Dies erfolgt mit dem Programm *DatAdd.exe*. Näheres dazu ist im Kapitel 6.2.1.1 zu finden.

Im zweiten Schritt müssen die Datenreihen gesichtet und gegebenenfalls bearbeitet werden. Dies ist notwendig, um die Rohdaten von Spikes und Sprüngen zu befreien. Im ungünstigsten Fall muss man „schlechte“, d.h. unbrauchbare Daten, per Hand löschen. Mit dem Programm *PreAnalyse.exe* lässt sich dies problemlos erledigen. Dieses Programm, das die meiste Entwicklungszeit benötigte, ist unter 6.2.1.2 näher beschrieben.

Der dritte Schritt der Datenaufbreitung umfasst das Anfügen der neuen Daten an bereits vorhandene. Für diesen Schritt wurde das Programm *EternaAdd.exe* entwickelt – Kapitel 6.2.1.3.

Nachdem die Daten nochmals mit *PreAnalyse.exe* kontrolliert wurden um eventuell noch vorhandene Fehler zu entfernen, folgt das Filtern der Daten. Das Programm *Filter.exe* ist so geschrieben worden, dass es verschiedene Filteroperatoren verwenden kann (s. a. Kapitel 6.2.2).

Um aus den gemessenen Werten die wahren Auslenkungen des Sensors zu erhalten, müssen die Daten kalibriert werden. Dies geschieht mit dem Programm *Kalibrierung.exe*. Neben dem Programm wird im Kapitel 6.2.3 auch die Ermittlung der Kalibrierfaktoren beschrieben.

Für viele Anwendungen müssen die Kanäle X und Y auf Nordsüd – Ostwest transformiert werden. Dies wird mit dem Programm *Transformation.exe* erledigt (im Detail in Kapitel 6.2.4).

Der Schritt „Ändern des Datenformates“ (Abb. 6.2) ist notwendig, da das folgende Programm zur Gezeitenanalyse ein spezielles Datenformat benötigt. Beschrieben ist dieser Unterpunkt unter 6.2.4.1.

Im Kapitel 6.2.5 wird die Analyse der Datenreihen mit dem Programm *Analyze.exe* (Wenzel, 1996) aus dem ETERNA-Programmsystem beschrieben.

Alle Programme bis auf das für die Analyse wurden in der Programmiersprache *Visual C# .NET* neu geschrieben. Einschränkend muss gesagt werden, dass dies nicht für jeden Schritt der Datenauswertung notwendig gewesen wäre, da bereits Programme vorhanden waren. Allerdings ermöglicht nun das neue Programmsystem eine homogene und komfortable Benutzung, so dass die Datenbearbeitung nun optimal und zügig durchgeführt werden kann. Die Programme lassen sich leicht mit Hilfe von übersichtlichen Batch-Dateien steuern. Ein weiterer Vorteil ist, dass im Falle unvorhergesehener Probleme, z.B. Fehler in der Datenauswertung, die einzelnen Schritte leicht manuell auszuführen sind, um den Fehler zu finden und zu eliminieren.

6.2.1.1 Programm – *DatAdd.exe*

Das Programm *DatAdd.exe* ist bereits die dritte Generation von Programmen, welche die Rohdatensätze zusammenfügen. Das Erste wurde unter der Sprache *Fortran77* (Jahr, 2005)¹ geschrieben. Man hat über die Konsole eingegeben, wie viele und welche Dateien in das ETERNA-Format konvertiert werden sollten. Zuvor war es aber nötig, die Dateinamen auf acht Stellen für den Namen und auf drei Stellen für die Extension zu ändern, da *Fortran* keine anderen Dateinamen akzeptiert. Das Zweite ist ein C-Programm, das in seiner Verwendung komfortabler aber noch nicht zufriedenstellend war.

Damit das Konvertieren der Dateien verständlicher wird, ist in der nachfolgenden Abb. 6.3 das Format der Rohdatei und der entsprechenden Datei im ETERNA-Format angegeben.

Die linke Seite der Abb. 6.3 zeigt das über das Internet erhaltene Rohdatenformat. Der Kopf der Datei, der acht Zeilen umfasst, enthält vier Kennungen, wobei jeweils zwei Zeilen zusammen gehören. In der jeweils ersten findet man die Beschreibung der Information der darunter liegenden Zeile. Die erste abrufbare Information ist die Abtastrate (*samplerate*). Sie beträgt hier 0.1 Hz, was einem Abtastintervall von 10 s

¹ Persönliche Mitteilung

entspricht. Die zweite Information, welche sich abrufen lässt, ist die Startzeit. Diese hier dargestellte Rohdatei beginnt am 01. Dezember 2003 um 0:28:57 Uhr. Die dritte und vierte Information sind der Stationsname und der betrachtete Kanal. Hier ist es die Station Berg und der X-Kanal. Die Information für den Kanal hat zwei Kennungen. Die Kennung „VAE“ beschreibt den X-Kanal und „VAN“ den Y-Kanal. Nach dem achtzeiligen Kopf folgen die gemessenen Werte in Volt.

SAMP_FREQ .100C00001 #START_TIME 2003 12 01 00 28 57.4762 #STATION_CODE BER #STATION_CHANNEL VAE 1.85836 1.85855 1.8584 1.85849 1.85869 1.85848 1.85868 1.85866 1.85867 1.85875 0 0 0 1.85878 1.85897 1.85874 1.85898	File : Berg X-Kanal Status : Askania Pendel KTB Start : 2003 12 01 00 29 00 End : Comment : Remarks : C***** ASK 1.000 77777777 .000 20031201 002900 1858.360 20031201 002910 1858.550 20031201 002920 1858.400 20031201 002930 1858.490 20031201 002940 1858.690 20031201 002950 1858.480 20031201 003000 1858.680 20031201 003010 1858.660 20031201 003020 1858.670 20031201 003030 1858.750 20031201 003040 99999.999 20031201 003050 99999.999 20031201 003100 99999.999 20031201 003110 1858.780 20031201 003120 1858.970 20031201 003130 1858.740 20031201 003140 1858.980
---	---

Abb. 6.3: Datenformate – links Rohdatensatz, rechts ETERNA-Format.

Das ETERNA-Format, dargestellt auf der rechten Seite der Abb. 6.3, besitzt ebenfalls einen Kopf, welcher aber 14 Zeilen lang ist. Ihm kann man die Station, den Kanal und die Startzeit entnehmen. Das Ausfüllen der Endzeit wird aus technischen Gründen unterlassen, da durch neuere Daten sich dies ständig ändern würde. Nach der Zeile, die mit einem „C“ beginnt auf das die Sterne folgen, beginnt der eigentliche Datensatz. Die Kennung „ASK“ besagt, dass die Daten von einem ASKANIA-Neigungsmesser stammen, und am Ende der Zeile die „1.000“ ist ein für die Programme des ETERNA-Paketes notwendiger Multiplikationsfaktor, welcher hier auf 1.0 gesetzt ist. Die folgende Zeile ist wiederum eine Kennung für die ETERNA-Programme. Die neu geschriebenen Programme verwenden diesen Kopf nicht, er wird deshalb nur kopiert oder mit Kommentaren ergänzt. Mit der 15. Zeile beginnt die Datenreihe. Die erste Zahlenspalte

ist das Datum in der Form Jahr-Monat-Tag. Die zweite Spalte gibt die Zeit in der Form Stunde-Minute-Sekunde an. In der dritten Spalte sind die Werte zu finden, welche im Vergleich zum ETERNA-Format mit 1000 multipliziert sind und demzufolge jetzt Millivolt bedeuten. Beim Vergleich der beiden Formate sind drei Zeilen auffällig (Abb. 6.3): Dies sind diejenigen, die eine mit „0“ bzw. „99999.999“ enthalten. Sie markieren eine Lücke in der Zeitreihe.

Das Programm ETERNA mit dem zugehörigen Format für die Daten wurde von Wenzel (1996) entwickelt und reagiert durch die Formate in den *Fortran*-Programmen sehr sensibel auf Fehler, wie zum Beispiel auf ein zusätzliches Leerzeichen am Ende der Zeile. In diesem Fall erscheint eine Fehlermeldung oder der Datensatz wird von dem Programm nur bis dahin bearbeitet. Bei der vorliegenden Arbeit wurde dieses Format verwendet. Dies war zwar programmiertechnisch etwas umständlicher, bietet aber den Vorteil, dass es hundertprozentig zwischen den neu entwickelten Programmen, den bereits existierenden Programmen und dem ETERNA-Programmsystem kompatibel ist.

Die erste Version dieses Programms war in der Anwendung sehr umständlich, da mit viel „editier“-Arbeiten behaftet. Bedingt durch die Menge der Daten erschien es notwendig, nach einer anderen Möglichkeit der Zusammenfügung der Rohdatensätze und deren Konvertierung in das ETERNA-Format zu suchen. Daraufhin ist die zweite Version entstanden. Dies ist ein *C-Programm*, das über eine Batch-Datei gesteuert wird. Mit Hilfe dieser Batch-Datei, welche eine Liste der in dem entsprechenden Verzeichnis befindlichen Dateien erstellt, kann sich das Programm die notwendigen Dateien anhand des Namens selbst heraus suchen. Danach hat das Programm jeweils entsprechend der Eingabeparameter wochenweise die Daten zusammengefügt und konvertiert. Die größte Schwierigkeit bei der Entwicklung dieses Programms war, dass es die zeitlichen Übergänge zwischen den einzelnen Rohdateien aufgrund fehlender oder doppelt besetzter Zeitkennung richtig vornimmt. Am Ende sollte die Zeitreihe von der Zeitkennung her vollständig sein. Die Tatsache, dass nur jeweils eine Woche zur weiteren Bearbeitung zusammengestellt wurde, beruht darauf, dass das ETERNA-Programm *Pregred.exe* (Wenzel, 1996), einem interaktiven grafischen Editor für digital erfasste Zeitreihen, immer wieder Probleme verursachte. Es war daher einfacher, nicht allzu lange Teilstücke zu bearbeiten. Aus den Unzulänglichkeiten von *Pregred.exe* ist die Idee gewachsen, ein neues Programm mit den gleichen Funktionen zu entwickeln. Nach funktionsfähiger Fertigstellung der neuen Variante – *PreAnalyse.exe* war es nicht mehr notwendig, die

Bearbeitung auf Datenreihen von einer Woche Länge zu begrenzen. Die im Internet vorhandenen ähnlichen graphischen Editoren (z.B. das Programm *TSoft*) wären sicherlich in der Lage gewesen das Problem mit *Pregred.exe* zu lösen. Aber durch die speziellen Anforderungen bei der Auswertung der Daten des KTB-Arrays war die Neuentwicklung sinnvoller. Mehr zu *Pregred.exe* und *PreAnalyse.exe* folgt im nächsten Kapitel.

Nachdem nun beliebig lange Datenreihen bearbeitet werden konnten, bot es sich an, eine neue Version des Programms, welche die Rohdaten konvertiert und zusammenfügt, zu entwickeln. Diese Version wurde *DatAdd.exe* genannt, was von „data addition“ abgeleitet ist. Der Vorteil dieses Programms ist, dass man die Anzahl der Tage und die Station und den Kanal frei über die Eingabeaufforderung wählen kann.



```

c:\ cmd

e:\ETERNA34\Auswertung-KTB>EternaAdd.exe

*****
* Das Programm benötigt mindestens 2 Eingabe Argumente! *
* EternaAdd [Quelldatei1] [Quelldatei2] *
* *
* Das Programm fuegt Dateien im ETERNA-Format zu einer Datei zusammen. *
* Die Samplerate wird aus den ersten beiden Werten der Quelldatei1 ermittelt. *
* Version 2.1 André Gebauer *
*****

e:\ETERNA34\Auswertung-KTB>

```

Abb. 6.4: Hilfe zu dem Programm *EternaAdd.exe*.

In Abb. 6.4 ist die Hilfe zu dem Programm *DatAdd.exe* zusehen, wenn man den Namen des Programms in die Eingabeaufforderung eingibt. Als Erstes verlangt das Programm die Kennung der Station, als zweiten Eingabeparameter den Kanal und danach die Start- und gegebenenfalls die Endzeit. In welcher Art und Weise dies zu geschehen hat, ist in der Abbildung zu sehen. Sicherlich ist richtig, dass das Programm sehr speziell auf das hier vorliegende Problem zugeschnitten ist, aber es ist sehr leicht zu modifizieren, um bei anderen ähnlich gearteten Problemen angewendet werden zu können. Dieses Programm wurde unter *Visual C#.NET* erstellt. Durch die Verwendung dieser Programmiersprache war es relativ einfach, das Programm zu entwickeln, da viele bereits vorhandene Objekte verwendet werden konnten. Dies gilt auch für die anderen neu entwickelten Programme.

Im gesamten Rahmen der Datenauswertung wird dieses Programm wieder durch eine Batch-Datei (Script im Anhang II.II) gesteuert, was vom Grundsatz her nicht nötig ist. So lässt sich das Programm effektiver nutzen, indem man einfach entsprechend der

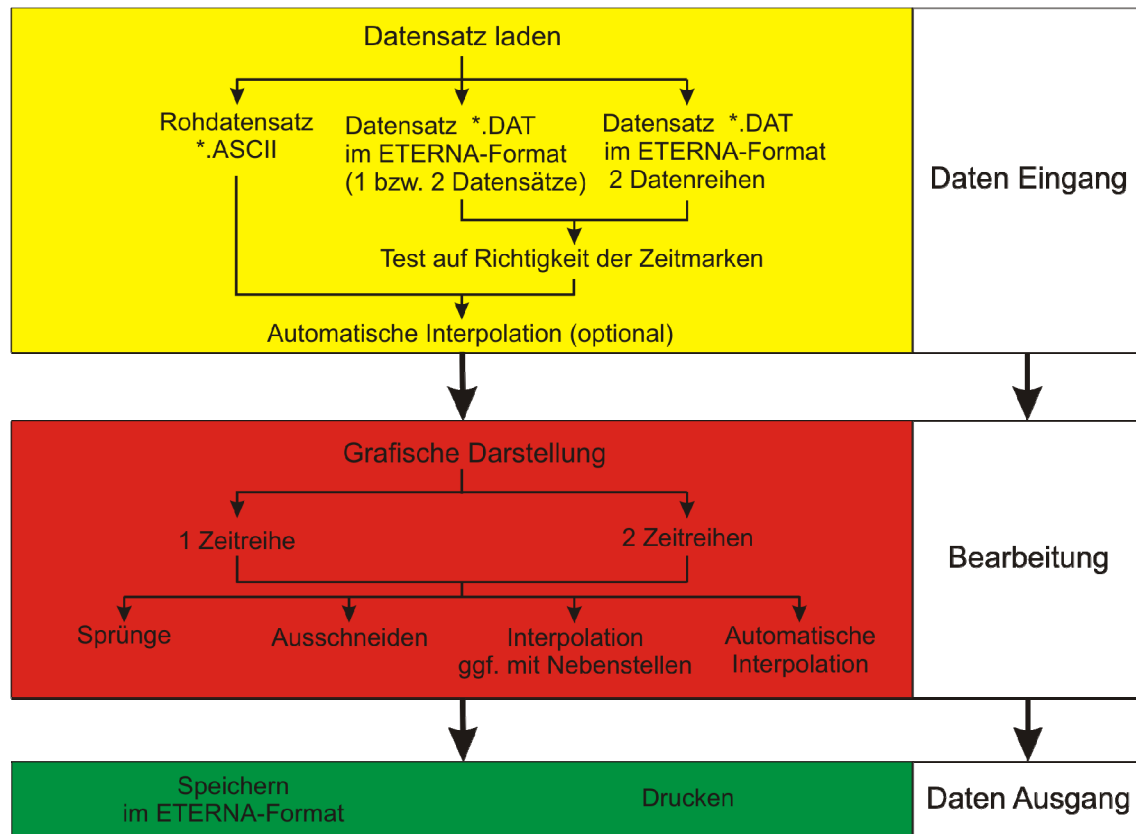
Eingabeparameter zum Beispiel alle Stationen mit den entsprechenden Kanälen bearbeiten kann. Der Programmcode für *DataAdd.exe* und die entsprechende Batchdatei mit demselben Namen sind im Anhang II.I zu finden.

6.2.1.2 Programm – *PreAnalyse.exe*

Das Programm *PreAnalyse.exe* ist eine Neu- bzw. Weiterentwicklung von *Pregred.exe* aus dem ETERNA-Programmsystem. Dabei wurden die Fähigkeiten von *Pregred.exe* optimiert, Unzulänglichkeiten aufgehoben und neue Funktionen entwickelt. Der auffälligste Unterschied ist, dass das neue Programm eine WINDOWS-Anwendung ist und nicht mehr umständlich über die DOS-Ebene gesteuert werden muss.

Pregred.exe war zu der Zeit, als es von Wenzel entwickelt wurde, sicherlich sehr fortschrittlich und nützlich. Aber im Laufe der Zeit haben sich die Rechner und Systeme weiter entwickelt, das Programm jedoch nicht. Um eine Datenreihe zu bearbeiten, ist es nötig, das Programm über eine modifizierte ini-Datei zu starten. Die Bearbeitung war nur mit der Tastatur möglich. Während der Bearbeitung von Datensätzen gab es häufig Probleme mit der richtigen Anzeige der Datenreihe, da der Anzeigebereich über die ini-Datei gesteuert wird und man nur hineinzoomen kann, aber nicht über den Ausgangszustand heraus. War der Wertebereich größer als das festgelegte Anzeigefenster, so war es sehr leicht möglich, den Überblick zu verlieren. Das größte Problem ergab sich aber aus einer Eigenart, die nicht erklärbar ist. Das Programm brach teilweise bei sehr stark gestörten Daten aber auch aus unerklärlichen Gründen zusammen. Um dennoch die Datenreihe zu bearbeiten, musste man diese kürzen und erneut bearbeiten. Dazu waren teilweise mehrere Anläufe nötig. Um von vornherein das Problem möglichst klein zu halten, wurden von *WocheAdd.exe* nur die Rohdaten wochenweise zusammengefügt und dann weiter bearbeitet. Nach den ersten Monaten der Bearbeitung der Daten stellte sich sehr schnell heraus, dass diese Arbeitsweise immer noch zu umständlich ist, was zur Entwicklung des Programms *PreAnalyse.exe* führte.

Nach anfänglichen programmiertechnischen Schwierigkeiten der ersten funktionsfähigen Version stellte sich dieses Programm als sehr nützlich, zuverlässig und ausbaufähig heraus. Während der ganzen Zeit der Datenbearbeitung wurde das Programm ständig durch neu auftretende Anforderungen weiterentwickelt. In Abb. 6.5 sind das Arbeitsschema und die verfügbaren Funktionen dargestellt.

Abb. 6.5: Schema der Fähigkeiten von *PreAnalyse.exe*.

Das Programm unterteilt sich in drei Hauptbereiche: die Dateneingabe, den Bearbeitungsbereich und die Datenausgabe. Doch bevor man die Datensätze laden kann, ist die Initialisierung der notwendigen Parameter für die Bearbeitung notwendig. Zu diesem Zweck gibt es eine ini-Datei, die allgemeine Parameter festlegt. In ihr stehen zum Beispiel die Samplerate und die Menge der Daten, die beim Laden eines Datensatzes im Arbeitsfenster angezeigt werden soll. Natürlich werden auch noch weitere Parameter geladen, aber diese können im gewissen Rahmen geändert werden. Wird aber während des Ladevorganges eines Datensatzes festgestellt, dass die Samplerate von der in der ini-Datei definierten Rate abweicht, ist es möglich, über ein Menü eine Auswahl zu treffen, wie *PreAnalyse.exe* verfahren soll.

Zum Laden eines Datensatzes öffnet man, wie bei WINDOWS-basierten Anwendungen gewohnt, die entsprechende Datei über den Standard „Öffnen“-Dialog. Dabei hat man die Möglichkeit, drei verschiedene Formate zu laden. Das erste Format, das man öffnen kann, war das bereits erwähnte ETERNA-Format. Nachdem nun aus den ersten beiden Datenzeilen die Samplerate ermittelt wurde, hat man bei Unstimmigkeiten

die Möglichkeit, diese zu ändern. Im dargestellten Problem wird die allgemein häufig benutzte Samplerate von 10 s verwendet. Während des Ladens der Datei wird natürlich die Samplerate kontinuierlich überprüft. Treten Fehler auf, so handelt das Programm selbstständig. Bei den Fehlern kann es sich darum handeln, dass entsprechende Zeitmarken fehlen. In diesem Fall werden diese einfach aufgefüllt. Natürlich könnte es auch sein, dass eine oder mehrere Zeitmarken doppelt auftreten. Dann werden diese entfernt. Sollte ein Fehler im Format auftreten, weil man zum Beispiel die Datenreihe per Hand bearbeitet und zufällig am Ende der Zeile ein Leerzeichen vergessen hat, so wird von der Laderoutine die Datenreihe nur bis dorthin geladen.

Die Rohdaten haben wie bereits beschrieben ein anderes Format. Aber um eine erste Kontrolle zu haben, ob das Messgerät richtig arbeitet, ohne die Rohdaten zu konvertieren, ist es sinnvoll, auch dieses Format öffnen zu können. Dabei wird, wie unter Abschnitt 6.2.1.1 erwähnt, die Startzeit der Datenreihe aus der Datei extrahiert, und danach werden fortlaufend die einzelnen Werte geladen.

Auf das dritte Format, das eine Variante des ETERNA-Formates ist, soll an späterer Stelle eingegangen werden.

Im Interesse eines schnellen Zugriffs auf die Daten bei der Bearbeitung wird die Datenreihe intern binär abgespeichert. Dies erfolgt kontinuierlich, indem eine Gleitkommazahl (Double) an die andere geschrieben wird. Sicherlich hätte man die gesamte Datenreihe auch in den RAM des Rechners laden können. Dies wäre aber nur sinnvoll gewesen, wenn man sich sicher ist, dass auf allen Rechnern, auf dem das Programm laufen soll, die Systemleistung ausreichend hoch ist. Von Anfang an bestand aber die Bedingung, dass dieses Programm auf allen WINDOWS-basierten Rechnern lauffähig sein soll.

Mit Hilfe der temporären Binärdatei lassen sich nun die Daten programmier-technisch sehr leicht und schnell modifizieren. Zu diesem Zweck müssen diese erst graphisch dargestellt werden. Beim ersten Anzeigen wird ein Wert aus den initialisierten Parametern herangezogen, wie viele Daten in dem Fenster angezeigt werden sollen. Daraufhin wird die entsprechende Anzahl der Daten nach ihren Maximalwerten durchsucht und nach entsprechender Skalierung für die Anzeige des gesamten Wertebereiches angezeigt. Zur graphischen Darstellung, die bei der Entwicklung des Programms die größten Schwierigkeiten bereitete, gehören zwei weitere hier nicht aufgeführte Funktionen. Das Scrollen durch den Datensatz, was entweder mit dem Mausrad erfol-

gen kann oder mit Hilfe der Bild nach oben bzw. – nach unten Tasten Das hinein und hinaus Zoomen in und aus dem Datensatz gehört natürlich auch zu den darstellenden Funktionen. Über den Zoomfaktor ist festgelegt, um wie viel Prozent sich der Anzeigebereich verändern soll. Dieser ist standardmäßig, wie auch die Scrollrate, auf 30% festgelegt. Diese Werte haben sich als relativ praktikabel erwiesen, können aber geändert werden.

Um nun Datenreihen editieren zu können, wurden zu diesem Zweck, die schon in *Pregred.exe* vorhandenen Funktionen Korrektur von Sprüngen, Ausschneiden von Daten und lineare Interpolation zwischen zwei Werten, eingebettet. Somit hatte *PreAnalyse.exe* dieselben Funktionen wie sein Vorgänger. Zusätzlich werden aber noch weitere Funktionen hinzugefügt. Dies war als erstes die automatische Interpolation.

Diese Funktion wurde entwickelt, weil sich bei der Bearbeitung der Daten herausstellte, dass in den Datensätzen ständig kleine Lücken von wenigen Werten, wie in Abb. 6.6 zu sehen ist, vorhanden waren. Startet der Benutzer diese Funktion, entweder über das Menü oder über die Tastenkombination „Strg+A“, so scannt die Routine den gesamten Datensatz ab und sucht nach Lückenkennungen und ermittelt die Anzahl der fehlenden Werte. Ist der ermittelte Wert unter einem Grenzwert, der auf 240, also sechs Minuten für 10 Sekundendaten festgelegt wurde, so wird eine lineare Interpolation vorgenommen (Jahr et al., 2005)¹. Der Grenzwert für die automatische Interpolation wird bei der Initialisierung festgelegt, kann aber auch über den Menüpunkt Einstellungen geändert werden. Ist das Ende des Datensatzes durch diese Routine erreicht, so wird die Anzeige wie nach jeder editierenden Funktion aktualisiert.

Die Editierung per Hand mit den drei schon angesprochenen Funktionen lässt sich wie unter *Pregred.exe* über die Tastatur erledigen. Dazu ist es nur nötig, den Cursor über die Pfeiltasten der Tastatur an die gewünschte Stelle zu bewegen und mit den entsprechenden Tasten die Funktionen auszulösen. Einfacher und schneller lässt sich dies mit Hilfe der Maus erledigen, wobei die Maustasten in dem aktiven Arbeitsbereich so einfach wie möglich belegt worden sind.

Die Funktion des Mausekorns wurde bereits erwähnt. Die linke Maustaste ist dabei dreifach belegt. Man muss sich aber als Benutzer keine Sorgen über die richtige Auswahl der entsprechenden Funktion machen, da es je nach Art und Weise der Verwendung automatisch gewählt wird. Ein einzelner Klick hat in Abhängigkeit von den

¹ Persönliche Mitteilung

Daten entweder die Funktion, eine Lücke mit Hilfe der linearen Interpolation zu füllen, falls dort, wo sich der Cursor befindet, eine Lücke ist. Die zweite Funktion bezieht sich darauf, dass sich dort keine Lücke sondern ein „echter“ Wert befindet. In diesem Fall wird die Funktion ausgelöst, um einen Sprung in den Daten zu korrigieren. Dazu wird die senkrechte Cursormarkierung fixiert und eine vertikale hinzugefügt. Mit dieser kann man nun die Sprunghöhe festlegen. Technisch wird dies so gelöst, dass über die Skalierung und der Anzahl der Pixel zwischen Ausgangspunkt und der auszuführenden Sprunghöhe ein Offset berechnet wird. Auch bei sehr großen Sprunghöhen ist dieses Verfahren sehr genau, da nach dem Auslösen der Sprungfunktion das Arbeitsfenster aktualisiert wird. Ist nun der Sprung nicht so wie es sich der Nutzer vorstellt, so lässt er sich korrigieren so oft es auch nötig ist.

Wird die linke Maustaste gedrückt gehalten und über den aktiven Arbeitsbereich gezogen, kann man die lineare Interpolation auslösen. Dabei wird der zu interpolierende Bereich rötlich transparent unterlegt. Die Funktion des Ausschneidens wird auf dieselbe Art und Weise ausgelöst, nur mit dem Unterschied, dass der Nutzer dazu die rechte Maustaste benutzen muss. Die vorhandenen Werte werden hier einfach mit der Lückenkennung belegt. Der Anfangs- und Endwert für die beiden Funktionen Interpolation und Ausschneiden werden wiederum über das Arbeitsfeld und die Cursorposition sowie die Menge der Anzahl der angezeigten Daten ermittelt.

In Abb. 6.5 ist die Funktion lineare Interpolation mit einem Zusatz belegt. Man hat als Benutzer die Möglichkeit die Art und Weise der Interpolation zu beeinflussen. Man kann in den Einstellungen wählen ob man diese mit oder ohne Einbeziehung der Nebenstellen vornehmen möchte. Die Normale ohne Nebenstellen berechnet den Anstieg der Interpolationsgeraden aus den beiden Endwerten. Wählt man aber die Funktion mit Nebenstellen aus, so wird eine Regression mit einer definierten Anzahl von Werten an den Rändern vorgenommen. Dabei wird der erste und der letzte zu interpolierende Wert extrapoliert und danach erst die Berechnung der Regressionsgerade vorgenommen. Diese aufgesetzte Funktion ist erst relativ spät entstanden. Der Grund dafür war, dass bei bestimmten Konstellationen der Werte, entweder einer verrauschten Datenspur oder dem Signal eines Erdbebens in dem Arbeitsbereich, die Interpolation unter Umständen nicht zufriedenstellend war. Am Besten ist dies in der folgenden Abbildung (6.6) zu sehen.

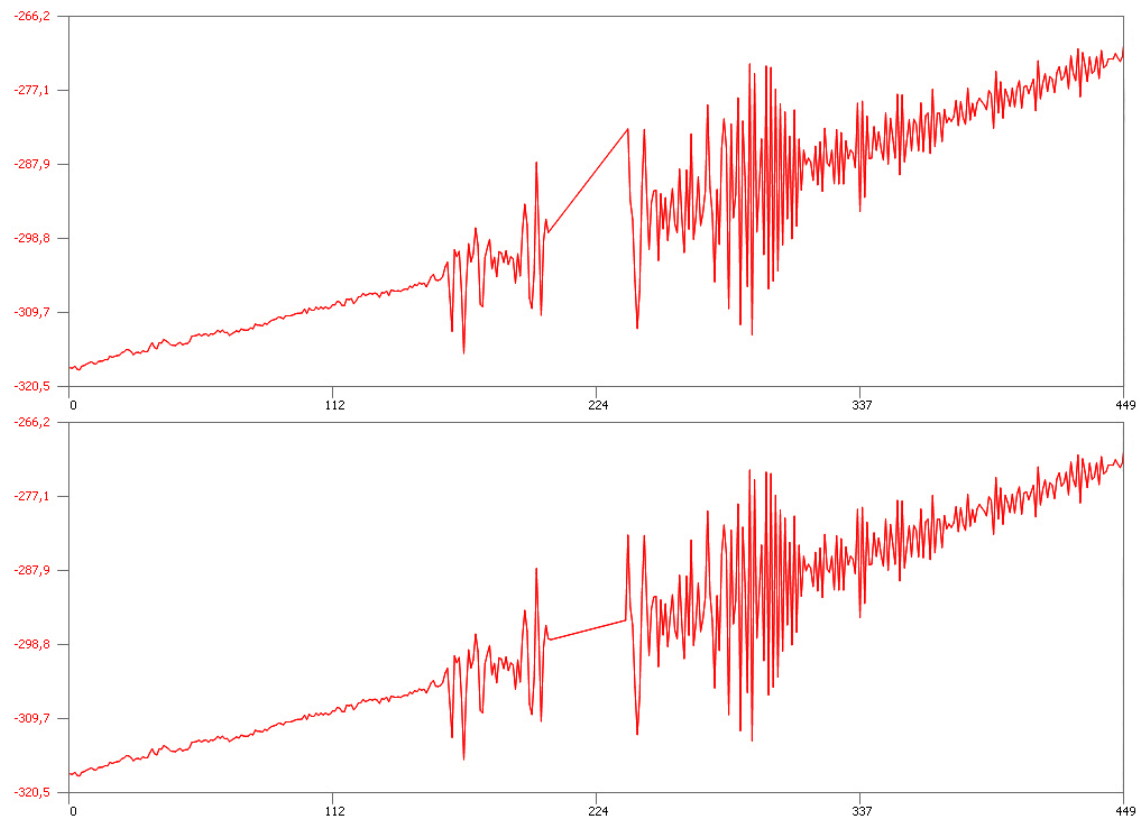


Abb. 6.6: Darstellung der Interpolation mit und ohne Nebenstellen.

Die Abbildung ist aus zwei Einzelabbildungen zusammengesetzt. In der oberen Hälfte ist die Auswirkung der linearen Interpolation ohne Nebenstellen zu sehen, darunter mit Nebenstellen. Um diese Abbildungen zu erstellen, wurden die gleichen Daten verwendet. Das jeweils wichtige Detail befindet sich in der Mitte der Graphen. Sehr deutlich ist die Extrapolation des letzten zu interpolierenden Wertes zu erkennen. Der Quellcode der oben beschriebenen Funktionen Sprungkorrektur, Ausschneiden von Werten und Interpolation ist im Anhang II.III abgedruckt.

Nachdem nun die Datenreihe editiert wurde, sollen die Daten weiter zur Verfügung stehen. Damit ist es notwendig die Datenreihe abzuspeichern. Eine Datenreihe kann mit *PreAnalyse.exe* nur im ETERNA-Format abgespeichert werden. Während des Ladens eines Datensatzes der Ausführung der automatischen Interpolation und des Speicherns werden in eine Report-Datei Eintragungen vorgenommen. Diesen Report anzusehen ist natürlich interaktiv möglich. Die Eintragungen beziehen sich im Fall des Ladens auf die Anzahl und möglichen fehlenden Datenzeilen. Für die automatische Interpolation werden die Zeiten und die Länge der interpolierten Lücke vermerkt. Bei der Speicherung sind die Zeitmarken und wiederum die Länge der Lücken eingetragen.

Weiterhin ist es möglich im Bereich der Datenausgabe die Datenreihe, wie sie in dem aktivem Arbeitsfenster zu sehen ist, zu drucken. Um das Druckbild modifizieren zu können, hat ein Benutzer im Menüpunkt Einstellungen die Möglichkeit, dem Plot einen Namen zu geben sowie die Achsen zu beschriften. Wie man dies vornehmen kann, ist der Abb. 6.7 (rechte Seite) zu entnehmen.

Alle unter *PreAnalyse.exe* möglichen Einstellungen und deren Darstellung im Arbeitsfenster ist in der nächsten Abb. 6.7 zu sehen.

In Abb. 6.7 auf der linken Seite ist ein Punkt „Header (Zeilenanzahl)“ zu finden. Dieser Parameter gibt an, wie viele Zeilen des Datensatzes zum Kopf gehören. Diese werden beim Laden eines Datensatzes gespeichert und beim Speichern als erstes wieder in die Ausgabedatei geschrieben. Die Parameter horizontale bzw. vertikale Hilfslinien kann man auswählen, um einen besseren Überblick zu erhalten. In dieser Arbeit wird diese Option aber nicht verwendet. Jedoch hat der nachstehende Wert eine aktive Bedeutung, auch wenn die Schaltflächen nicht aktiviert sind. Sie geben die Anzahl der Tickmarks an, welche in der aktiven Arbeitsfläche angezeigt werden sollen. Der letzte Parameter „Größe der Bewegung des Cursors durch die Pfeiltasten“ gibt die Sprungweite des Cursors bei Benutzung der Pfeiltasten zur Editierung an. Unten links in dem Einstellungsfenster befindet sich die Schaltfläche „Standard“. Sie dient dazu, Einstellungen zurückzusetzen, falls diese geändert worden sind. Dabei werden genau dieselben Einstellungen hergestellt, die auch gerade zu sehen sind.

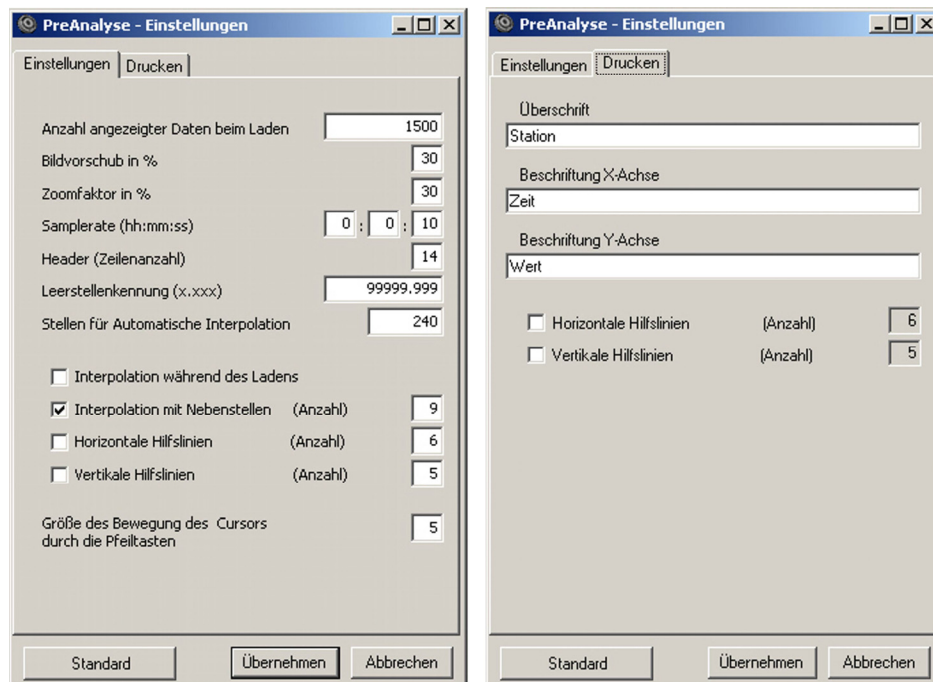


Abb. 6.7: Möglichkeiten der Einstellung unter *PreAnalyse.exe*.

Bevor die Oberfläche von *PreAnalyse.exe* beschrieben wird, seien noch zwei weitere Funktionen erläutert. Wenn man Änderungen in Dateien vornimmt, ist man mit dem Ergebnis nicht immer zufrieden und möchte diese rückgängig machen. Gegenüber seinem Vorgänger besitzt *PreAnalyse.exe* diese Funktion. Dies ist aber nur einmalig möglich. Weiterhin hat man die Wahl, zwei Datensätze gleichzeitig zu laden und zu bearbeiten. Das Laden kann auf zwei Wegen geschehen: Zum einen kann man zwei Dateien im unter 6.2.1.1 beschriebenen ETERNA-Format laden. Dabei werden anhand der Zeitkennungen die Datenreihen zeitlich richtig übereinander dargestellt. Zum anderen lässt sich ein Datensatz mit zwei Zeitreihen (in einer Datei) laden, der im ETERNA-Format vorliegt. Sind nun zwei Datenreihen geladen, so will man als Nutzer diese auch parallel bearbeiten. Dabei kann man jede Datenreihe für sich alleine oder aber auch beide gemeinsam mit den schon beschriebenen Funktionen editieren. Die Möglichkeit, beide Datenreihen gemeinsam zu bearbeiten, wurde als letztes in das Programm integriert, da die Idee erst relativ spät entstand. Diese Möglichkeit erleichtert aber enorm die Arbeit und spart vor allen Dingen sehr viel Zeit, da von einer Station beide Kanäle meistens an derselben Stelle editiert werden müssen.

Die Oberfläche von *PreAnalyse.exe* umfasst folgende Funktionen (Abb. 6.8): Das WINDOWS-Standardfenster beinhaltet ein Menü, eine Statuszeile, die aktive Ar-

beitsfläche und einen Bereich, um schnell die Anzeigeeigenschaften der Arbeitsfläche zu ändern. Dieser Bereich befindet sich unter dem Menü und besteht aus vier Elementen: Eine Scrollleiste, um eine beliebige Position in der bzw. den Datenreihen anzuwählen, einen Schiebeschalter, um die Breite zu beeinflussen, und eine Schaltfläche, um die gewählten Einstellungen auszuführen. Wahlweise kann man mit der Schaltfläche „Auto Update“ auch seine Einstellungen synchron übertragen lassen. Bei großen Datensätzen kann dies unter Umständen sehr lange dauern.

Mit der Statuszeile kann man Informationen über die Position des Cursors im Bezug auf den aktiven Datensatz erhalten. Das erste Feld gibt die Zeitkennung zurück und das zweite enthält eine Angabe für den laufenden Index. Das Dritte gibt den Wert der aktiven Datenreihe zurück, an der sich der Cursor befindet.

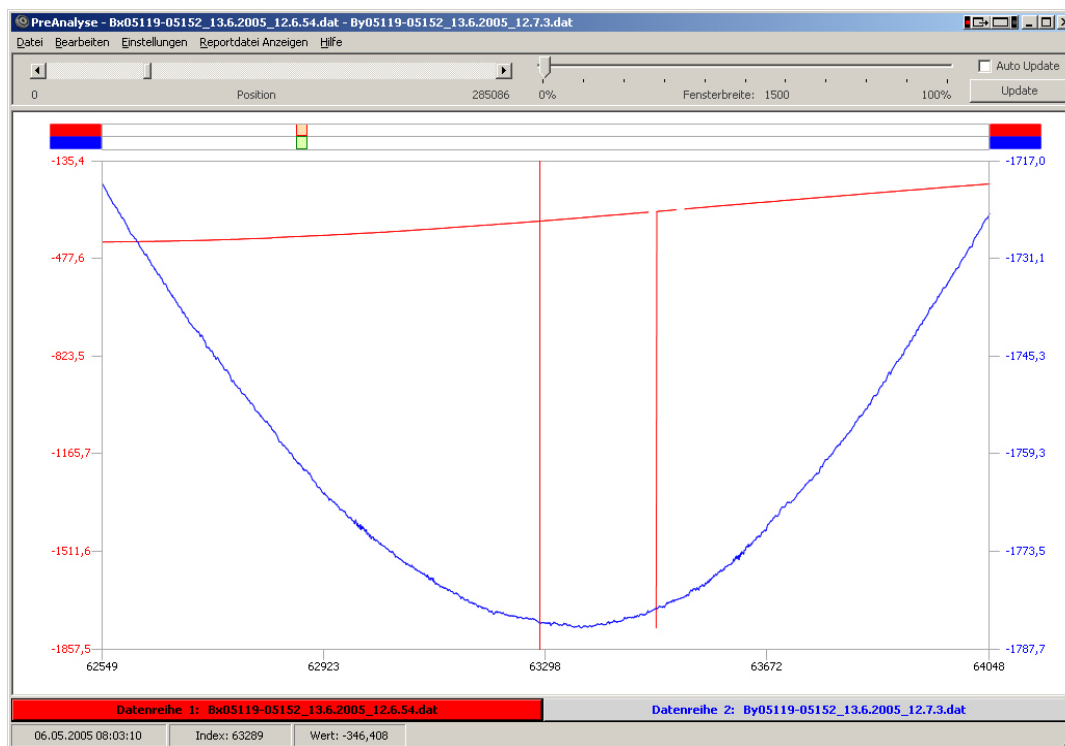


Abb. 6.8: Das Programm *PreAnalyse.exe* mit einem geladenen Beispiel-Datensatz.

Der Arbeitsbereich besteht im Fall, dass zwei Datenreihen geladen sind, aus drei Elementen, sonst nur aus einem. Über zwei Schaltflächen, die sich im unteren Bereich des Fensters befinden, kann der Nutzer zwischen den beiden Datenreihen hin und her wechseln. Dieses ist auch mit der Tastatur möglich, wozu die Tasten „1“, „2“ bzw. „0“ zu benutzen sind. „1“ und „2“ erklären sich von selbst und „0“ dient dazu, beide Daten-

reihen anzuwählen. Welche Datenreihe aktiv ist oder ob beide ausgewählt sind, zeigen wiederum die Schaltflächen an. In dem in Abb. 6.8 gezeigten Beispiel ist die erste Datenreihe aktiv. Zu erkennen ist das an der roten Hinterlegung der Schaltfläche. Wäre die zweite aktiv, so wäre auch die zweite Schaltfläche farbig hinterlegt, aber mit blau. Für den Fall, dass beide aktiv sind, wären beide Schaltflächen farbig hinterlegt.

Die Arbeitsfläche besteht aus einem aktiven und einem anzeigenden Bereich. Der aktive Bereich ist eingerahmt von den Achsen des Diagramms. Die Abszisse gibt den laufenden Index der Werte der Datenreihe an. Die Ordinaten rechts bzw. links geben einen Anhalt, in welchem Wertebereich sich die angezeigten Datenreihen befinden. Die Zuordnung erfolgt wiederum farbig. Im oberen Teil befindet sich eine Anzeigeleiste, welcher man mehrere Informationen entnehmen kann. Die gesamte Länge beschreibt die Länge der Datenreihen relativ, die beiden kleinen Fenster (rot und grün) geben die Fensterbreite und die Position der Reihen zur gesamten Datenreihe wieder. Der farbliche Unterschied der Fenster geht auf die Datenreihen selbst zurück. Der blaue Graph, die zweite Datenreihe, enthält keine Lücken, und das wird mit dem grünen Fenster angezeigt. An ihm ist auch keine Editierung mehr vorzunehmen. Im Roten sind zwei kleinere Lücken zu erkennen, daher auch das rote Fenster. Die zwei kleinen Lücken im Graphen lassen sich durch die schon beschriebene Interpolation beheben, wobei man bei der ersten Lücke etwas mehr Arbeit investieren muss, da sich an ihrem Ende ein Spike befindet. Entweder geht man über den Weg, dass man den Spike durch die Funktion „Ausschneiden“ entfernt und danach einfach durch einmaliges Drücken der linken Maustaste im Bereich der Lücke interpoliert. Oder man benutzt das Aufziehen eines Fensters mit der linken Maustaste über die Lücke hinweg, wobei man aber unter Umständen Datenwerte ändert, die eigentlich korrekt sind.

Aufgrund der Komplexität wurde das Programm von Wissenschaftlern, die mit ähnlichen Datensätzen umgehen, getestet. Aufgrund dieser Tests wurden Fehler weitestgehend beseitigt. Das Programm wurde nicht nur für die Tiltmeter-Daten genutzt, sondern auch für die Aufbereitung der Grundwasser- und Wetterdaten, wobei gerade die Funktionen mit ihren besonderen Eigenheiten, wie der Test und Korrektur auf die richtige Zeitkennung während des Ladens eines Datensatzes, sehr nützlich waren.

6.2.1.3 Programm – *EternaAdd.exe*

Nachdem nun die Zeitreihen frei von Fehlern sind, müssen die neuen Reihen an die vorhandenen angefügt werden. Dies kann mit Hilfe des Programms *EternaAdd.exe* vorgenommen werden. Dieses Programm existiert in der zweiten Version. Die erste Version hatte den Namen *WocheAdd.exe*. Entstanden ist es im Zusammenhang mit der ersten Version der Datenauswertung als die Programme *WocheDat.exe* und *Pregred.exe* benutzt wurden. Im Zuge der weiteren Entwicklung hat es sich angeboten dieses Programm, das unter *C* geschrieben worden war, neu zu entwickeln. Die Gründe dafür waren, dass es für den Ablauf der Datenbearbeitung einfacher ist, wenn alle benutzten Programme unter der gleichen Programmiersprache geschrieben sind. Der zweite Aspekt war, dass sich aus den Erfahrungen mit der Arbeit mit den Daten die Rahmenbedingungen, die im Programm gesetzt waren, um die Datenreihen zusammenzufügen, geändert hatten.

EternaAdd.exe prüft als erstes, ob die Zeitmarken in den Dateien richtig sind und fügt sie dann anhand der Rahmenbedingungen zusammen. Die Samplerate wird aus den ersten beiden Datenzeilen der ersten Datenreihe bestimmt. Dies ist nicht sehr vorteilhaft, falls dort ein Fehler vorhanden ist. Aber unter der Annahme, dass die Dateireihe schon öfter verwendet wurde und demnach zumindest im Bezug auf die ersten Zeitmarken bzw. Samplerate richtig ist, ist diese Vereinfachung in der Benutzung nicht als Nachteil anzusehen. Sind die Zeitmarken richtig, so wird die letzte Zeile der ersten Datei und die erste Zeile der Zweiten dazu benutzt, um zu ermitteln, wie das Programm verfahren soll, um den Übergang nahtlos erfolgen zu lassen.

Das Programm entscheidet anhand der bestimmten Samplerate, wie der Übergang erfolgen soll. Ist die Samplerate verschieden von 10 Sekunden, so werden die Daten ohne Offset zusammengefügt und dabei gibt es drei Möglichkeiten, wie sich die zwei Zeitmarken zueinander verhalten können. Die erste ist, dass es laut Samplerate zwei folgende Zeitmarken sind. Im zweiten Fall fehlen zwischen den beiden Dateien Zeitmarken, dann werden sie einfach mit der Lückenkennung aufgefüllt. Der letzte Fall ist, dass sich die Datenreihen zeitlich überlappen. Dann werden solange die Zeilen der zweiten Datenreihe überlesen, bis sie nahtlos ineinander übergehen.

Ist die Samplerate 10 Sekunden, so werden die drei Fälle weiter unterteilt. Dabei werden der erste und der letzte Fall gleich behandelt, wobei nur die letzte Zeile der ersten Datenreihe und die entsprechende folgende Zeitmarke der zweiten Datenreihe im

Bezug zur ersten betrachtet werden. Die feinere Unterteilung bezieht sich dann auf die Werte selber, genauer auf die Differenz der beiden. Im Detail ist dies in der Tabelle 6.1 dargestellt. Die Werte und Grenzen basieren allesamt auf Erfahrungen. Prinzipiell sollte man aber wissen, wie ein Offset zwischen den einzelnen Datenreihen entstehen kann. Bei Überschreitung des Messbereichs, z.B. durch starke Driften, müssen die Messbereiche manuell nachgestellt werden. Dies liegt aber in einer Größenordnung, die weit außerhalb der angegebenen Grenzen liegt. Der verbleibende Fall, dass zwischen den Datenreihen Werte fehlen, ist etwas schwieriger. Dabei wird die Anzahl der fehlenden Zeitmarken mitbetrachtet. Explizit ist dies wieder der Tabelle 6.1 zu entnehmen.

Tab. 6.1: Vorgehensweise in *EternaAdd.exe* im Fall von 10-Sekunden-Daten um anhand der ermittelten Parameter der beiden Datenreihen diese zusammenzufügen.

aufeinander folgende Zeitenmarken / überlappende Zeitenmarken		Zeitenmarken fehlen		
Differenz der Werte kleiner 0.5 mV	Offset ist null	Anzahl fehlender Stellen größer 15	Zeitmarken werden mit Leerstellen aufge- füllt	
Differenz der Werte größer 0.5 mV	Offset wird berech- net aus Differenz der vorhandenen Wer- te	Anzahl fehlender Stellen kleiner 15	Differenz der Werte kleiner 4.0	Werte werden linear interpoliert
			Differenz der Werte größer 4.0	Werte werden mit Leerstellen aufgefüllt

Nachdem nun die Zeitreihen erfolgreich zusammengefügt worden sind, ist es ratsam, nach den hier dargelegten Randbedingungen die nun längere Datenreihe mindestens an der Nahtstelle, wie in Abbildung 6.2 unter Schritt 1.4 angedeutet ist, visuell mit *PreAnalyse.exe* zu sichten und notfalls zu korrigieren. Dann steht die Zeitreihe für den nächsten Schritt in der Datenauswertung, der Filterung auf Stundenwerte, zur Verfügung.

Um die Konsolenanwendung zu starten, sind wie in Abb. 6.9 zu sehen, zwei Eingabeparameter notwendig. Nach dem Namen der auszuführenden Datei wird der Name der Datei der ersten Datenreihe erwartet, an welche die zweite Datei, die als zweites Argument erwartet wird, anzufügen ist. Während des Testens der ersten Datei wird ein Backup mit der Extension *.bak angelegt, damit im Fall eines Fehlers der Datensatz erhalten bleibt. Die aus den zwei Datenreihen zusammengesetzte Datei hat denselben Namen wie die erste Quelldatei. Der Quellcode von *EternaAdd.exe* ist in Auszügen im Anhang II.IV zu finden.

```

cmd
e:\ETERNA34\Auswertung-KTB>EternaAdd.exe

*****
* Das Programm benötigt mindestens 2 Eingabe Argumente! *
* *
* EternaAdd [Quelldatei1] [Quelldatei2] *
* *
* Das Programm fuegt Dateien im ETERNA-Format zu einer Datei zusammen. *
* Die Samplerate wird aus den ersten beiden Werten der Quelldatei1 ermittelt. *
* *
* Version 2.1 André Gebauer *
*****

e:\ETERNA34\Auswertung-KTB>

```

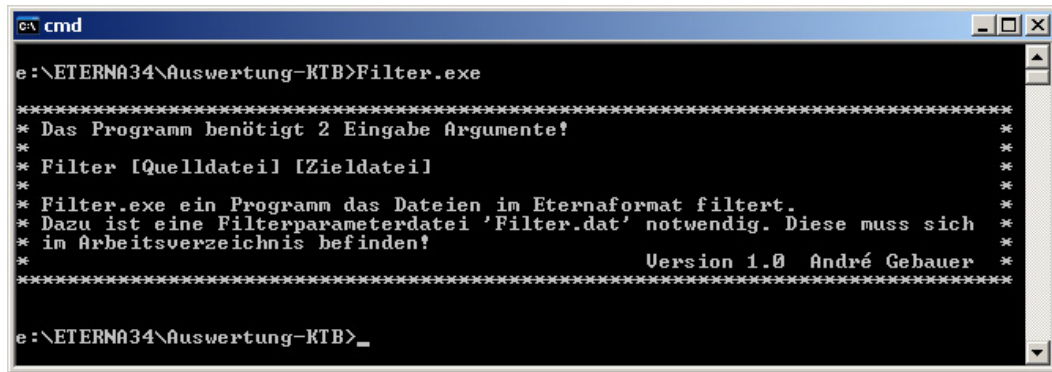
Abb. 6.9: Konsolenhilfe von *EternaAdd.exe*.

6.2.2 Filtern der Datenreihen – *Filter.exe*

Das Programm *Filter.exe* war nicht notwendiger Weise neu zu schreiben gewesen, allerdings ist die neue Anwendung deutlich komfortabler. Der Vorgänger war ein *Fortran* Programm, das fest vorgegebene Dateinamen als Input verlangte. Demnach war es immer nötig, den zu filternden Datensatz umzubenennen. Die Ausgabedatei hatte einen ähnlichen Namen wie die Eingabedatei, was eine unnötige und zusätzliche Fehlerquelle darstellte. Weiterhin war es nie klar, wenn man das Programm angewendet hatte, ob es wirklich richtig arbeitete oder nicht, da man als Benutzer keine Rückmeldung bekam. Zusätzlich benötigte das Programm zu dem Filteroperator noch eine ini-Datei. In der neuen Version wurde dieses Problem so gelöst, dass die benötigten Parameter der ini-Datei in dem Kopf der Datei stehen, in welcher sich auch der Filteroperator befindet.

Die Abb. 6.10 enthält die Hilfe des Programms *Filter.exe*. Ihr ist zu entnehmen, dass zwei Eingabeargumente erforderlich sind; zusätzlich muss sich der Filteroperator im selben Verzeichnis befinden, in dem man die Quelldatei filtern möchte. Das Programm gibt zwei Dateien aus, die bis auf die Extension den gleichen Namen haben.

Das Programm benötigt zwei Eingabeargumente, die zu filternde Quelldatei im ETERNA-Format und den Namen der Zieldatei. Zusätzlich benötigt es die Datei, in der die Initialisierungsparameter stehen, und den Filteroperator, welcher standardmäßig den Namen *Filter.dat* hat. Diese Datei muss sich in dem Arbeitsverzeichnis befinden, in welchem auch der zu filternde Datensatz steht.



```
CA cmd
e:\ETERNA34\Auswertung-KTB>Filter.exe
*****
* Das Programm benötigt 2 Eingabe Argumente! *
* Filter [Quelldatei] [Zielfile] *
* *
* Filter.exe ein Programm das Dateien im Eternaformat filtert. *
* Dazu ist eine Filterparameterdatei 'Filter.dat' notwendig. Diese muss sich *
* im Arbeitsverzeichnis befinden! *
* Version 1.0 André Gebauer *
*****
e:\ETERNA34\Auswertung-KTB>_
```

Abb. 6.10: Konsolenhilfe von *Filter.exe*.

Die Datei *Filter.dat* enthält, wie schon angedeutet, zwei Teile, den Kopf der zur Initialisierung des Filterprozesses notwendig ist und die Werte des Filteroperators, welche mit einem laufenden Index versehen sind. Einen Auszug aus der Datei ist in Abb. 6.11 zu finden. Der Kopf der Datei enthält für *Filter.exe* wichtige Informationen. Die erste Zeile ist ein Kommentar. In der Zweiten steht die Anzahl der Stützstellen. Während des Ladens dieser Datei wird diese Angabe verwendet um zu testen, ob der Filteroperator auch vollständig ist. Die folgende Zeile gibt an, wie groß der Abstand der Stützstellen der gefilterten Zeitreihe sein soll. Da für jede Stunde ein Wert gewünscht wird, ist der Parameter „1“. Die Zeile „Anzahl der Filterelemente“ ist Kommentar. Die fünf Zeilen, die auf die Zeile „Filterparameter“ folgen, erklären sich von selbst. Die Zeilen bis dorthin, wo die Zahlenreihen beginnen, sind wiederum nur Kommentar. Der Filteroperator besteht aus zwei Spalten. In der ersten befindet sich der laufende Index, welcher nach dem Einlesen mit dem Parameter Anzahl der Stützstellen verglichen wird. Die zweite Spalte enthält den Operator.

Nachdem nun der Filteroperator und die notwendigen Parameter bekannt sind, wird aus dem Datensatz die erste Datenzeile gelesen und nach Zeitkennung und dem dazugehörigen Wert ausgewertet. Die Zeitkennung wird darauf getestet, dass sich entsprechend der Samplerate der Zielfile, hier eine Stunde, auch zu einer vollen Stunde beginnt. Falls dies nicht der Fall sein sollte, so wird so lange mit den folgenden Zeilen getestet, bis die geforderte Bedingung erfüllt ist.

```

LINEARER DATENUNABHAENGIGER FILTER
ANZAHL DER STUETZSTELLEN : 12241
STUETZSTELLENABSTAND      : 1
ANZAHL DER FILTERELEMENTE: 2
HANNING FENSTER

FILTERPARAMETER:
Faktor mit dem Zeitreihe multipliziert wird           : 1.0
Stuetzstellenabstand der zu filternden Daten in Sekunden : 10
Stuetzstellenabstand der gefilterten Daten in Sekunden   : 3600
Anteil der Zeitreihe die zu Beginn u. Ende nicht ausgegeben werden in%: 10.0
Lueckenkennung                                           : 99999.999

I NR   I   TYP      I      GRENZFREQUENZ (EN) IN CPH      I
I      I      I      I  UNTERE GRENZFREQ. I OBERE GRENZFREQ. I
I-----I-----I-----I-----I-----I-----I-----I
I  1   I  TIEFPASS  I      I      I      I      I      I
I  2   I  FLANKE (TP) I      C.4000 I      I      I      I
I-----I-----I-----I-----I-----I-----I-----I

1 C.0000C300000000E+0C
2 -C.21877624824206E-12
3 -C.50227426060210E-12
4 -C.9086C529673013E-12
5 -C.14408144316247E-11
6 -C.21003329703590E-11
*
*
*
12237 -C.14408144316247E-11
12238 -C.9086C529673013E-12
12239 -C.50227426060210E-12
12240 -C.21877624824206E-12
12241 C.0000C300000000E+0C

```

Abb. 6.11: Auszug aus der Datei *Filter.dat* (10 Sekundenwerte auf Stundenwerte).

Der Filterprozess beginnt nun damit, dass die Hälfte der Stützstellenanzahl des Filteroperators aus der Datenreihe eingelesen wird. Diese wird dann gespiegelt, damit man die volle Anzahl der Stützstellen nutzen kann und nicht so viele Werte verliert. Daraufhin wird der Stundenwert berechnet. Für den zweiten Stundenwert wird der Operator um eine Stunde verschoben. Dies geht so weiter, bis in der Datenreihe eine Lückenkenung gefunden wird oder das Ende der Datenreihe erreicht ist. Geschieht dies, so werden die Daten der Zeitreihe wieder gespiegelt. War es nicht das Ende der Datei, so beginnt der ganze Prozess von vorn. Durch die Spiegelung der Daten an den Enden der tatsächlichen Zeitreihe werden nicht vorhandene Daten simuliert, um die vorhandenen Lücken nicht noch größer werden zu lassen. Deshalb muss man berücksichtigen, trotz des Kunstgriffes der Spiegelung, dass man Daten verliert. Dies wird durch den vorletzten Initialisierungsparameter geregelt. In Abhängigkeit von der Länge des Filteroperators in Stunden werden die angegebenen Prozente an Stundenwerten auf die Lückenkenung gesetzt.

Der Filter, der von 10-Sekunden-Daten auf Stunden-Daten filtert, ist ein transversaler symmetrischer Tiefpassfilter, der aus 12241 Stützstellen besteht, über welchen

ein Hanningfenster gelegt ist. Er ist für Frequenzen unterhalb von 0.1 mHz durchlässig und oberhalb von 0.125 mHz völlig gesperrt (Abb. 6.12). Die Länge des Filteroperators von 34 Stunden ist auf die scharfe Frequenzcharakteristik zurückzuführen.

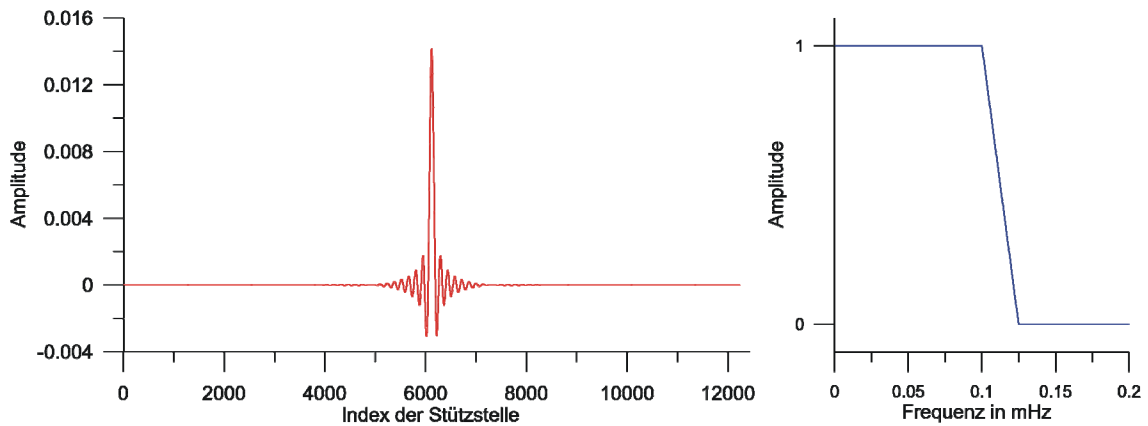


Abb. 6.12: Filteroperator für Filterung von 10 Sekundendaten auf Stundendaten mit 12241 Stützstellen (links) und Frequenzcharakteristik (rechts - idealisiert).

Das Programm *Filter.exe* lässt sich natürlich auch mit anderen Filteroperatoren benutzen, was im Zusammenhang mit der Aufbereitung der Grundwasserdaten auch notwendig ist. Die Filterdateien haben dabei das bereits vorgestellte Format. Der Quellcode des Programms *Filter.exe* befindet sich im Anhang II.V.

6.2.3 Kalibrierfaktoren – Kalibrieren der Datenreihen

Damit die gemessenen Daten der einzelnen Stationen vergleichbar sind, müssen die Zeitreihen kalibriert werden. Dieser Vorgang erzeugt aus den gemessenen Werten in Millivolt die wahren Neigungen in Millisekunden.

6.2.3.1 Bestimmung der Kalibrierfaktoren

Die Neigungsmesser besitzen, wie schon in Kapitel 5.1.2 beschrieben, eine in situ Kalibrierung. Wie sich diese Kalibriersprünge in den Daten widerspiegeln, ist in Abb. 6.13 zu sehen. Die Höhe der Sprünge beträgt rund 300 mV. Die Schalteinheit löst wöchentlich 10 Sprünge im Abstand von 10 Minuten pro Kanal aus (nach Manual). Zunächst wird der X-Kanal kalibriert, und nach einer Pause von 10 Minuten der Y-Kanal. Im gezeigten Beispiel (X-Kanal der Station Berg) sieht man das Übersprechen des Y-

Kanals. Dieses Übersprechen ist darauf zurückzuführen, dass die beiden Kanäle nicht ganz senkrecht aufeinander stehen (Weise, 1992).

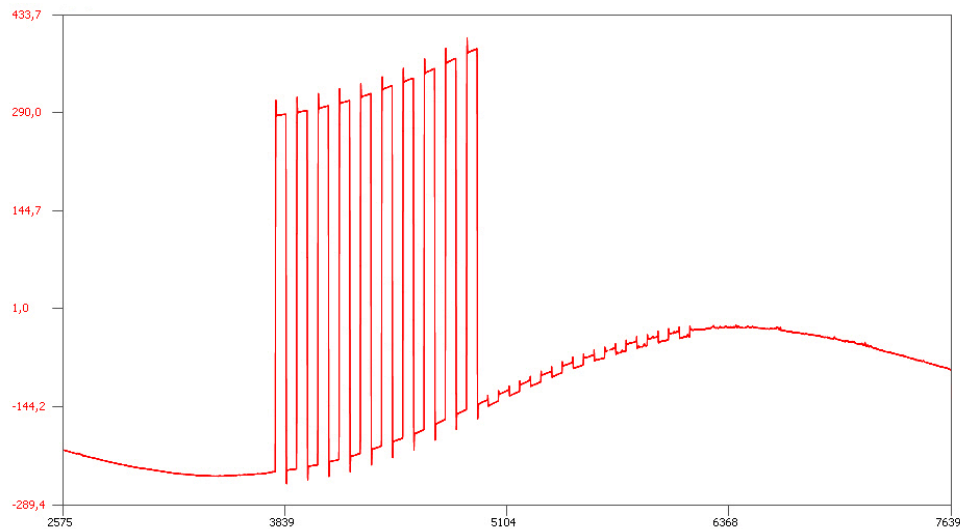


Abb. 6.13: Kalibriersprünge (Ausschnitt aus einer Zeitreihe von der Station Berg – Darstellung aus *PreAnalyse.exe*; Y-Achse in mV).

Zur Auswertung der Sprünge wurden *Fortran*-Programme verwendet (Weise, 1992). Dabei wird die Höhe der Sprünge über numerische Verfahren ermittelt. Dazu ist es ratsam, mehr als nur 10 Sprünge zu verwenden. Im Mittel wurden etwa 1000 Sprünge verwendet. Um nun aus den gemittelten Werten der Sprunghöhen (in Millivolt) die wahre Neigung zu ermitteln, wird diese nach Gleichung 5.2 unter Zuhilfenahme des Winkelausschlages (Tab. 5.2) berechnet. In Tabelle 6.2 sind die Ergebnisse dargestellt. Zusätzlich sind die Zeiträume eingetragen, für die die Faktoren gültig sind, da sie bei einem Wechsel des Messsystems neu bestimmt werden müssen. Zudem ist in der Tabelle die Identifikationsnummer des Instrumentes enthalten.

Tab. 6.2: Kalibrierfaktoren der Stationen.

Station	Gültiger Zeitraum		Instrument ID	Kalibrierwinkel X 10^{-3} sec	X - Komponente in ms/mV	Kalibrierwinkel Y 10^{-3} sec	Y - Komponente in ms/mV
	Von	bis					
Berg	02.09.'03	21.10.'04	103	50.01	0.1282406	49.41	0.1460288
Berg	21.10.'04	-	103	50.01	0.0944915	49.41	0.0941302
Eiglasdorf	15.10.'03	-	111	33.70	0.1659167	33.30	0.1384425
Mittelberg	02.09.'03	-	112	31.70	0.1664654	32.10	0.1470918
Püllersreuth	21.07.'03	-	102	20.63	0.0644176	20.63	0.0587863
Stockau	01.08.'03	31.08.'04	108	30.78	0.1274160	31.19	0.1211013
Stockau	31.08.'04	-	104	49.80	0.1684008	46.55	0.1488842

6.2.3.2 Kalibrieren der Daten – *Kalibrierung.exe*

Das Programm *Kalibrierung.exe* ist ein einfaches Programm. Es multipliziert die Werte der entsprechenden Zeitreihe mit dem jeweiligen Kalibrierwert. Die Steuerung des Programms erfolgt über Parameter. Dazu werden von dem Programm, wie in Abbildung 6.14 dargestellt, drei Eingabeparameter erwartet: Der Name der Quelldatei, die Zieldatei und der entsprechende Kalibrierwert.



```

c:\ cmd
e:\ETERNÄ34\Auswertung-KTB>Kalibrierung.exe
*****
* Das Programm benötigt 3 Eingabe Argumente! *
* Kalibrierung [Quelldatei] [Zieldatei] [Kalibrierwert] *
* Kalibrierung.exe ein Programm das Dateien im Eternaformat den Wert mit dem *
* Kalibrierwert multipliziert! *
* Version 1.0 André Gebauer *
*****
e:\ETERNÄ34\Auswertung-KTB>

```

Abb. 6.14: Konsolenhilfe von *Kalibrierung.exe*.

Das Programm liest aus der Quelldatei, nachdem der Kopf der Datenreihe kopiert wurde, die Zeilen der Datenreihe ein, extrahiert den Wert der dritten Spalte (siehe Abschnitt 6.2.1.1) und multipliziert ihn mit dem Kalibrierfaktor. Dies erfolgt aber nur, solange der Wert nicht die Lückenkennung ist. Die neue Zeile wird in die Zieldatei geschrieben und der Prozess beginnt von vorn, solange, bis das Ende der Daten erreicht wurde. Der Quellcode zu *Kalibrierung.exe* ist im Anhang II.VI zu finden.

6.2.4 Transformieren der Datenreihen – *Transformation.exe*

Es wurde schon im Kapitel 5.1.3 erwähnt, dass die Orientierung der Komponenten der Tiltmeter nicht mit den Himmelsrichtungen Nord-Süd und Ost-West identisch ist. Um nun die Werte der einzelnen Stationen vergleichen und sinnvoll auswerten zu können, müssen die X- und Y-Kanäle erst einmal auf Nordsüd und Ostwest transformiert werden. Dieses Kapitel orientiert sich an Graupner (2001). In Abb. 6.15 ist der geometrische Zusammenhang des Problems dargestellt. Zur Lösung des Problems der Transformation wird im Folgenden eine kurze Herleitung der zur Transformation nötigen Gleichungen gegeben.

Der Azimutwinkel zwischen dem Y- Kanal und der Nord-Richtung ist mit α bezeichnet. Die rote Ebene in Abb. 6.15 bestimmt den Neigungswinkel in X- und Y-Richtung. Bezeichnet man den Neigungswinkel in X-Richtung mit χ und den in Y-Richtung mit ψ , so lautet die Gleichung für die Ebene

$$-(\tan \chi) \cdot x - (\tan \psi) \cdot y + z = 0 \quad (6.1)$$

Alle Koordinatenachsen, X- und Y-Achse und Nord- und Ost-Achse, bis auf die Z-Achse liegen in einer Ebene und werden somit als Koordinatenebene bezeichnet. Die Punkte K und G liegen auf der Neigungsebene, deren Lotpunkte L und H auf der Koordinatenebene liegen. Die Abstände der Lotpunkte seien 1 vom Koordinatenursprung aus gesehen. Im X-Y-Koordinatensystem haben dann die Punkte K bzw. G die gleichen Koordinaten wie L bzw. H .

$$\begin{aligned} x_K &= \sin \alpha & y_K &= \cos \alpha \\ x_G &= -\cos \alpha & y_G &= \sin \alpha \end{aligned} \quad (6.2)$$

Mit Gleichung (6.1) berechnen sich die Z-Koordinaten der beiden Punkte K und G folgendermaßen:

$$\begin{aligned} z_K &= \tan \chi \sin \alpha + \tan \psi \cos \alpha \\ z_G &= -\tan \chi \cos \alpha + \tan \psi \sin \alpha \end{aligned} \quad (6.3)$$

Mit diesen Koordinaten lassen sich jetzt der Neigungswinkel η in N-Richtung und der Neigungswinkel ε in E-Richtung berechnen

$$\begin{aligned} \eta &= \arctan z_K = \arctan(\tan \chi \sin \alpha + \tan \psi \cos \alpha) \\ \varepsilon &= \arctan z_G = \arctan(-\tan \chi \cos \alpha + \tan \psi \sin \alpha) \end{aligned} \quad (6.4)$$

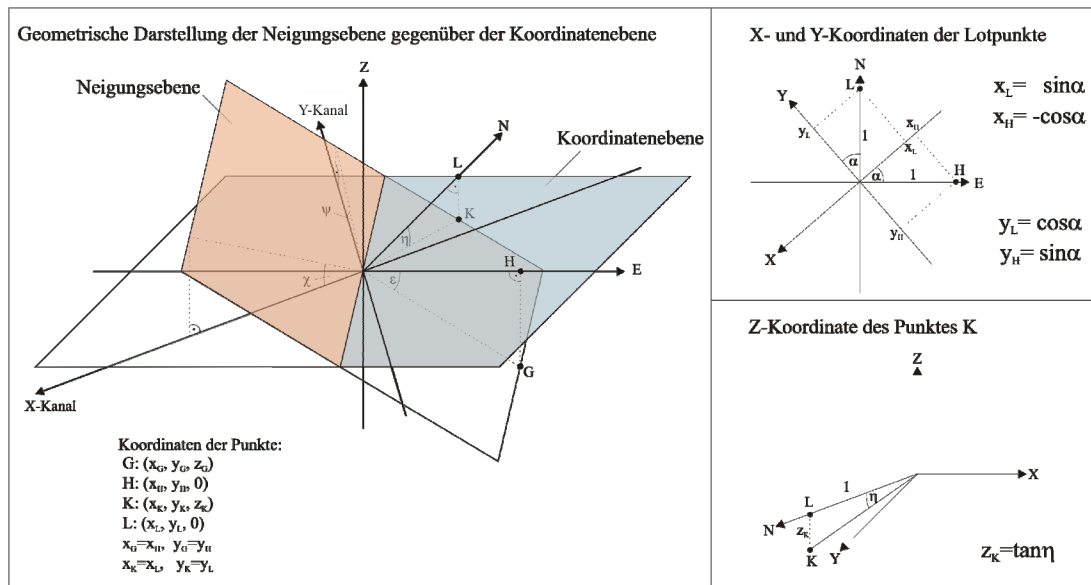


Abb. 6.15: Transformation der Daten auf Nordsüd und Ostwest.

Die Gleichungen (6.4) in einer anderen Form geschrieben heißen

$$\begin{aligned} EW &= -x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \\ NS &= x \cdot \sin(\alpha) + y \cdot \cos(\alpha) \end{aligned} \quad (6.5)$$

Die Variablen x und y in den Gleichungen werden aus den Datenreihen der jeweiligen Kanäle gewonnen. Die Variable α geht als Eingabeargument wie die Namen der Quell- und Zieldateien in das Programm *Transformation.exe* ein (siehe Abbildung 6.16).

```

cmd
e:\ETERN34\Auswertung-KTB>Transformation.exe
*****
* Das Programm benötigt 5 Eingabe Argumente!
*
* Transformation.exe [1] [2] [3] [4] [5]
* [1] Quelldatei X-Kanal 'KanalX.dat'
* [2] Quelldatei Y-Kanal 'KanalX.dat'
* [3] Zieldatei NS-Kanal 'KanalNS.dat'
* [4] Zieldatei EW-Kanal 'KanalEW.dat'
* [5] Azimut 'xxx.xx'
*
* Version 1.0 André Gebauer
*****
e:\ETERN34\Auswertung-KTB>

```

Abb. 6.16: Hilfe zur Nutzung von *Transformation.exe*.

Die Variable α wird über die Orientierung des Einbaus der Tiltmeter bestimmt (s. a. Kapitel 5.1.3). Die Azimute wurden sowohl optisch als auch seismisch bestimmt

(Jahr et al., 2005). In der folgenden Tabelle 6.3 sind die ermittelten Azimutwinkel zusammengestellt, welche für die Transformation verwendet werden.

Tab. 6.3: Azimute für Transformation der Kanäle.

Station	Gültiger Zeitraum		Instrument ID	Transformationswinkel des Y-Kanals in Richtung Nord in °	Transformationswinkel des Y-Kanals in Richtung KTB in °
	von	bis			
Berg	02.09.'03	-	103	135.8	218.0
Eiglasdorf	15.10.'03	-	111	138.0	282.1
Mittelberg	02.09.'03	-	112	310.0	211.0
Püllersreuth	21.07.'03	-	102	345.0	352.3
Stockau	01.08.'03	31.08.'04	108	172.0	29.9
Stockau	31.08.'04	-	104	331.0	188.9

Die Transformation erfolgt mit den Gleichungen (6.5) und den dargestellten Parametern. Die Berechnung der transformierten Werte *EW* und *NS* erfolgt nur dann, wenn *X* bzw. *Y* keine Lückenkennung ist.

Ein Programm, das zur Transformation der Datenreihen benutzt werden kann, war vorhanden. Es war nur leider etwas umständlich, dies zu verwenden, da wie bei allen schon vorgestellten *Fortran* – Programmen es nicht möglich war, dem Programm Eingabeargumente zu übergeben. So ist dann dieses Programm entstanden. Der Quellcode ist im Anhang II.VII zu finden.

6.2.4.1 Ändern des Datenformates – *ChFormat.exe*

Das Ändern des Datenformates mit *ChFormat.exe* ist kein echter Schritt im Prozess der Datenauswertung, aber notwendig für die Gezeitenanalyse, die im nächsten Schritt beschrieben wird. Der Name ist abgeleitet von „change format“. Es werden bei der Änderung nur die Werte, die die Lückenkennung enthalten, gegen eine Kennung ersetzt, die das Programm *Analyze.exe* aus dem ETERNA-Programmsystem braucht. Am Ende der Datei wird noch eine weitere zusätzliche Kennung benötigt. Die Änderungen, die vorgenommen werden, sind vergleichend in der folgenden Abb. 6.17 zu sehen. Das Programm zum Ändern des Formates sucht einfach nach den entsprechenden Markierungen und ersetzt oder ergänzt die benötigten Kennungen.

20040523 090000 -427.223	20040523 090000 -427.223
20040523 100000 -417.513	20040523 100000 -417.513
20040523 110000 -428.846	20040523 110000 -428.846
20040523 120000 -458.932	20040523 120000 -458.932
20040523 130000 99999.999	99999999
20040523 140000 99999.999	EW Ask111 0.100 1.000
20040523 150000 99999.999	77777777 0.000
20040523 160000 99999.999	20040523 200000 -616.005
20040523 170000 99999.999	20040523 210000 -584.244
20040523 180000 99999.999	20040523 220000 -556.266
20040523 190000 99999.999	20040523 230000 -538.145
20040523 200000 -616.005	*
20040523 210000 -584.244	*
20040523 220000 -556.266	*
20040523 230000 -538.145	
*	20050630 190000 5693.780
*	20050630 200000 5687.351
*	20050630 210000 5672.310
20050630 190000 5693.780	99999999
20050630 200000 5687.351	88888888
20050630 210000 5672.310	

Abb. 6.17: Änderung der Formate im Vergleich.

Zur Anwendung des Programms sind wie in Abb. 6.18 angegeben drei Argumente nötig: die Quelldatei, der Name der Zieldatei und ein Kommentar, der sich auf den Kanal bezieht.

```

e:\ETERNA34\Auswertung-KTB>ChFormat.exe

*****
* Das Programm benötigt 3 Eingabe Argumente!
*
* ChFormat [Quelldatei] [Zieldatei] [Kanal:EW / NS]
*
* ChFormat.exe ein Programm das Dateien im Eternaformat aendert.
* Die Leerstellen werden durch eine Kennung für das Programm 'Eterna/Analyze'
* versehen!
*
* Version 1.0 André Gebauer
*****
e:\ETERNA34\Auswertung-KTB>

```

Abb. 6.18: Konsolenhilfe von *Chformat.exe*.

6.2.5 Gezeitenanalyse – *Analyze.exe*

Zur Analyse der Gezeiten-Information wird das Programm *Analyze.exe* aus dem ETERNA-Programmsystem verwendet (Wenzel, 1996).

Nach Wenzel (1997b) besteht das Ziel der Erdgezeitenanalyse darin, Informationen über die Reaktion des Systems ERDE-STATION-SENSOR auf die genau bekannten Gezeitenkräfte zu erhalten (Abb. 6.19). Die Eigenschaften des Sensors werden durch Kalibrierung (Abschnitt 6.2.3) bestimmt.

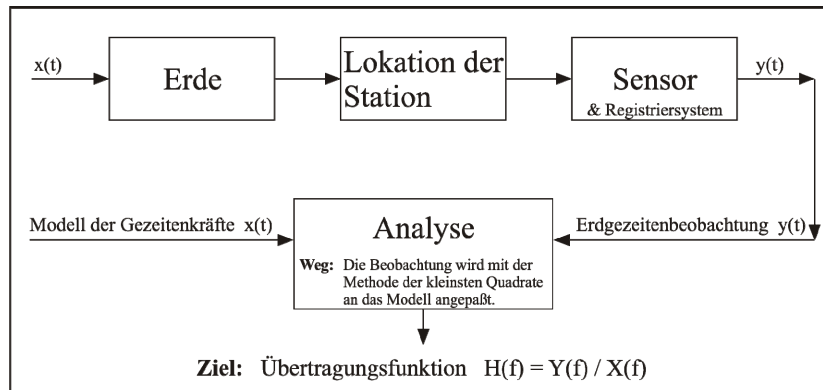


Abbildung 6.19: Funktionsweise und Ziel der Erdgezeitenanalyse (nach Wenzel, 1997b).

Für die Analyse von Erdgezeiten kommen weltweit verschiedene Programme mit teils unterschiedlichen Grundkonzepten zum Einsatz: HYCON–MC von Schüller (1986), das Programm von Venedikov (Vendikov, 1966; De Toro et al., 1993), BAY-TAP–G von Tamura et al. (1991) und als derzeit internationaler Standard ETERNA 3.30/3.40 von Wenzel (1997b). Dabei ist ETERNA ein ganzes Programmpaket für die komplette Bearbeitung von Zeitreihen, bei dem das Programm *Analyze.exe* die eigentliche Gezeitenanalyse ausführt (Wenzel, 1996).

Das Grundprinzip von ETERNA 3.30 beruht auf den von Chojnicki (1973) entwickelten Konzepten basierend auf der Methode der kleinsten Quadrate. Die Reaktion des Systems ERDE–STATION–SENSOR auf die gezeitenerzeugende Kraft wird durch ein lineares kausales Einkanalssystem beschrieben. Die „Inputfunktion“ $x(t)$ in Abb. 6.19 folgt aus der Gezeitenpotentialentwicklung und die „Outputfunktion“ $y(t)$ aus der Erdgezeitenbeobachtung. Daraus kann die Beobachtungsgleichung aufgestellt werden, in der die unbekannten Parameter Amplitudenfaktoren H und Phasenverschiebungen $\Delta\Phi$ mit der Methode der kleinsten Quadrate abgeschätzt werden (Wenzel, 1997b). Für eine konkrete Gezeitenanalyse berechnet *Analyze.exe* also die theoretischen Erdgezeiten $x(t)$ für eine bestimmte Station mit gegebenen geographischen Koordinaten und das Erdmodell (gleichmäßig rotierendes Ellipsoid mit flüssigem äußeren Kern und viskosem Mantel) von Dehant (1987) und passt die beobachteten Gezeitensignale $y(t)$ an die theoretischen Erdgezeiten $x(t)$ an. Das Programm folgt in der Syntax (siehe Anhang III) dem Standardformat der „Working Group on High Precision Tidal Data Processing“ (Jentzsch, 2005)¹ und ermöglicht die Analyse von Gezeiten für die Komponenten

¹ Persönliche Mitteilung

Schwere, Neigung, Volumen-, Vertikal- und Horizontalstrain (Wenzel, 1996). Für alle Komponenten werden die Amplituden, Amplitudenfaktoren, Phasenverschiebungen und die jeweiligen Standardabweichungen ausgegeben. Nach Aussagen des Programmautors ist *Analyze.exe* für die Analyse von Neigungsgezeiten weniger gründlich getestet worden als für die Schwere.

Die Methode der kleinsten Quadrate eignet sich deshalb so gut für die Gezeitenanalyse, weil die Ausgleichung auch bei Lücken in der Zeitreihe funktioniert, da sie Residuen zwischen Modell und Beobachtung liefert, die bei bekannten Frequenzen eine gezielte Entfernung grober Fehler ermöglichen. Zudem werden Standardabweichungen der ausgeglichenen Parameter bestimmt (Wenzel, 1997b). Die berechneten und in dieser Arbeit verwendeten Standardabweichungen sind erfahrungsgemäß zu gering geschätzt, da keine Korrelationen der Beobachtungen untereinander berücksichtigt werden (Wenzel, 1997b). Realistische Fehlerabschätzungen unter Verwendung von Fourier-Spektren der Residuen ergeben, je nach Datenqualität und Struktur der Lücken, etwa dreimal so große Fehler.

Eine Problematik der Erdgezeitenanalyse liegt in langperiodischen Driften, z. B. durch Instrumentengang, tektonische Bewegungen, meteorologische Einflüsse, oder durch den durch die Injektion veränderten Porendruck. Deshalb ist vor der Ausgleichsrechnung eine Driftreduktion notwendig, wobei sich zwei Methoden zur Unterdrückung, bzw. Separation der Drift anbieten. Zum einen kann eine Driftfunktion (in ETERNA mit Tschebyscheff-Polynomen) approximiert werden, und zum anderen lassen sich alle langperiodischen Signalanteile durch digitale Hochpassfilterung entfernen. Driftfunktionen werden üblicherweise nur für geringe Driften, beispielsweise bei der Analyse von Registrierungen supraleitender Gravimeter, aufgestellt (Wenzel, 1997b). Für die Driftunterdrückung der hier durchgeführten Analysen wurde der 51-Stunden Bandpassfilter von Pertsev (1959) verwendet. Abb. 6.20 zeigt die Charakteristik dieses Filters. Die Grenzfrequenz liegt ungefähr bei zwei Tagen. Alle Signale mit längeren Perioden werden herausgefiltert, so dass nur Gezeitenperioden von weniger als zwei Tagen verbleiben. Der Pertsev-Hochpassfilter betont zudem die ganz-, halb- und drittel-tägigen Frequenzen. Der Charakteristik des Filters ist zu entnehmen, dass der Übertragungsbereich keine scharfe Grenze hat und dass die Übertragung nicht bei 1 liegt, so dass hier noch Korrekturen angebracht werden müssen (vergl. auch Jentzsch, 1976).

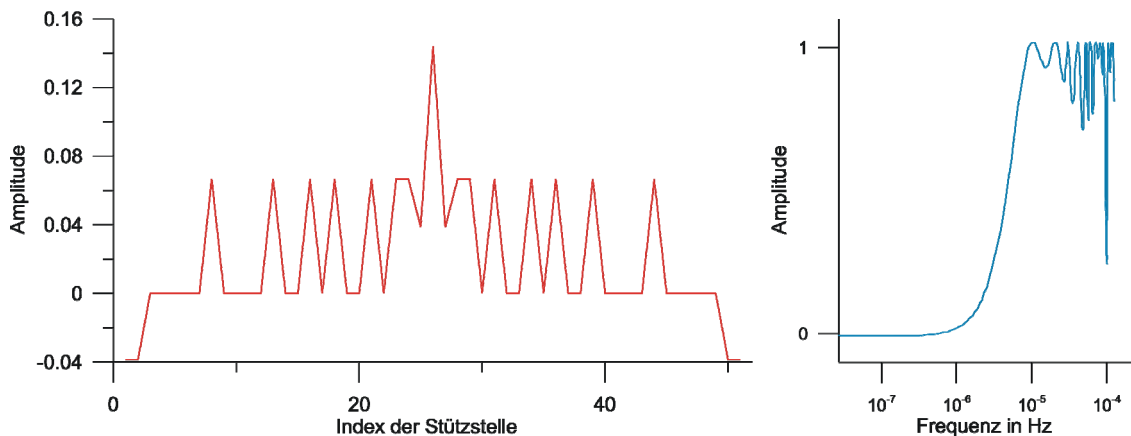


Abbildung 6.20: 51-Stunden Pertsev-Hochpassfilter (nach Pertsev, 1959).

Unter *Analyze.exe* besteht die Möglichkeit, den herausgefilterten Anteil auszugeben, was für das hier vorliegende Problem sinnvoll ist. Weiterhin sei darauf hingewiesen, dass von ETERNA eine neuere Version existiert, in der auch einige Änderungen der Fehlerabschätzung in dem Programm *Analyze.exe* angebracht wurden. Diese Version wurde aber nicht verwendet, da die beschriebene Fehlerabschätzung auf einem anderen Weg vorgenommen wird.

6.3 Darstellen des Datenauswertungssystems

Durch die große Menge an Daten der fünf Stationen ist es notwendig, zur optimalen Bearbeitung und Auswertung, diese in einer geeigneten Art und Weise zu verwalten. Dazu wurde ein komplexes System von Verzeichnissen angelegt. Die folgenden dargestellten Verzeichnisse und Unterverzeichnisse haben das Wurzelverzeichnis *\ETERNA34* (Abb. 6.21). Das verwendete Analyseprogramm *Analyze.exe* aus dem Programmsystem ETERNA 3.3 benötigt zur Arbeit die von ETERNA angelegte Verzeichnisstruktur und wird anstatt der neueren Version verwendet, da die Fehlerbetrachtung in der älteren Version besser in der Anwendung ist. Es werden dort die Fehler aus dem Fourier-Spektrum verwendet.

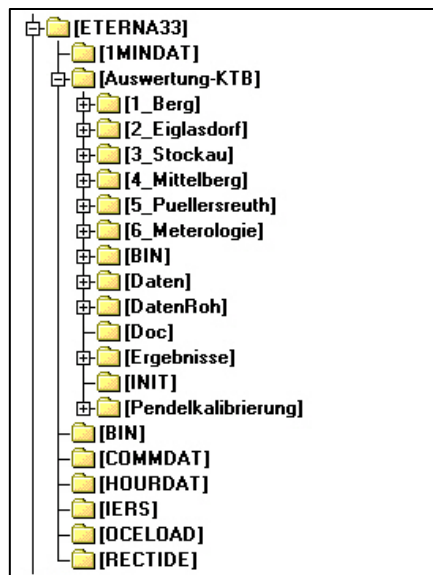


Abb. 6.21: Verzeichnisbaum der Auswertung der Tiltmeterdaten.

Das Wurzelverzeichnis des neuen Aufbereitungssystems *\Auswertung-KTB* zur Auswertung beinhaltet Unterverzeichnisse, die jeweils eine spezielle Aufgabe haben. In den sechs mit einer vorangestellten Zahl bezeichneten Verzeichnissen (*\1_Berg*, *\2_Eiglasdorf*, ...) befinden sich die jeweiligen aufbereiteten und analysierten Daten. In dem Verzeichnis *\BIN* sind die ganzen schon teilweise unter 6.2 beschriebenen Programme vorhanden. Das Verzeichnis ist als Path-Variable (unter WINDOWS) gesetzt, um auf die darin enthaltenen Programme zugreifen zu können. Das Verzeichnis *\Daten* ist das

Arbeitsverzeichnis, um die unter *\DatenRoh* liegenden Rohdaten zu bearbeiten. Unter *\Doc* liegen nützliche Textdokumente, die bei der Aufbereitung und Auswertung helfen können. *\Ergebnisse* enthält Ergebnisse und Plots, welche mit Grapher erstellt wurden. In dem Verzeichnis *\INIT* befinden sich die für das Programm *Analyze.exe* des ETERNA-Programmsystems notwendige ini-Dateien. Und das letzte Unterverzeichnis *\Pendelkalibrierung* ist das Arbeitsverzeichnis, um die unter 6.2.3 beschriebenen Kalibrierwerte zu ermitteln.

Die Unterverzeichnisse *\DatenRoh* und *\Daten* enthalten weitere Unterverzeichnisse. Die Daten, die als gepackte Dateien via Internet empfangen werden, sind unter

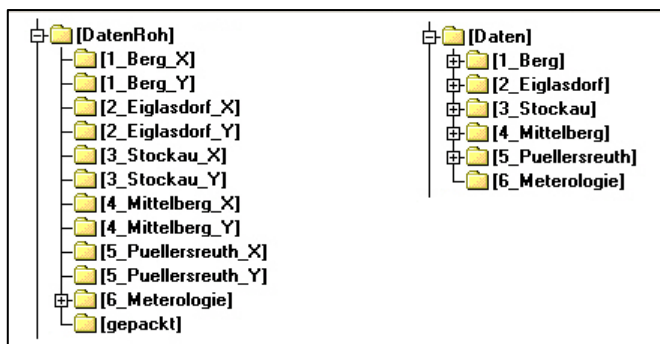


Abb. 6.22: Verzeichnisbäume *DatenRoh* (links) und *Daten* (rechts) unter „Auswertung-KTB“.

\gepackt abgelegt. Sind neue Teildatensätze vorhanden, werden diese nach *\DatenRoh* entpackt und werden mit Hilfe einer Batch-Datei in die entsprechenden Unterverzeichnisse der Kanäle der Stationen kopiert. Der nächste Schritt ist das unter

6.2.1.1 beschriebene Zusammenfügen und Konvertieren der Tagesrohdateien. Dies erfolgt zweckmäßiger Weise für alle Stationen gleichzeitig wiederum mit Hilfe einer Batch-Datei, welche auch die Datenreihen in die entsprechenden Unterverzeichnisse geordnet nach den Stationen nach \Daten (Abb. 6.22) kopiert. Danach erfolgt das Aufbereiten der Stücke mit *PreAnalyse.exe* (6.2.1.3), gefolgt vom Zusammenfügen der alten und neuen Reihen mit *EternaAdd.exe*. Natürlich ist es dann notwendig, die neuen längeren Datenreihen zu sichten. Die letzten drei Schritte müssen für die einzelnen Kanäle der Stationen per Hand ausgeführt werden, um wirklich sicher zu sein, dass die Daten vollständig und richtig aufbereitet wurden.

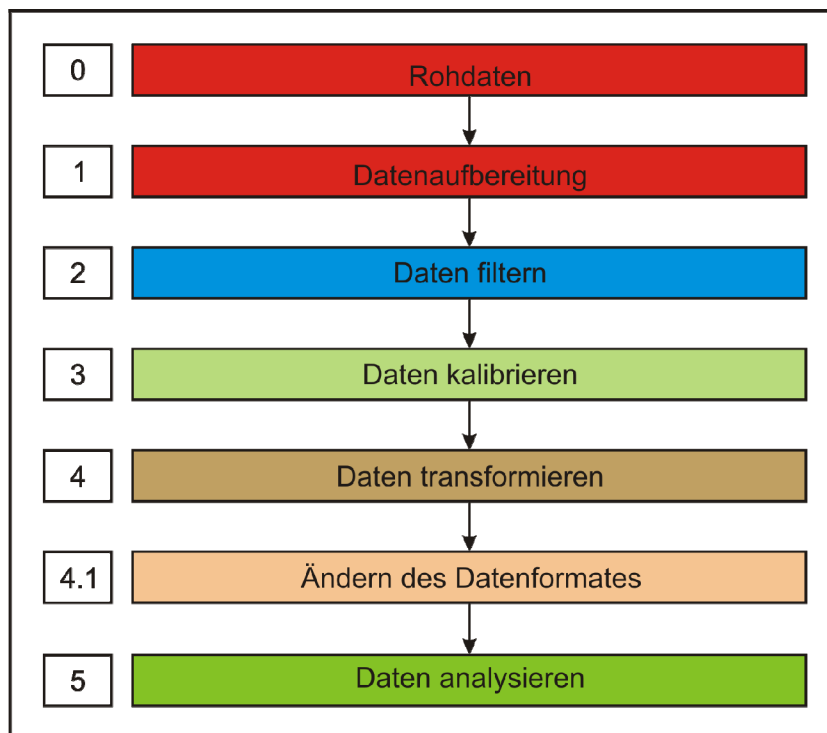


Abb. 6.23: Schema der Datenauswertung.

Die eben beschriebenen Schritte der Datenauswertung sind die Schritte „0“ und „1“ der Abb. 6.23. Nachdem man die längeren Zeitreihen mit Hilfe einer weiteren Batch-Datei in die Unterverzeichnisse \Daten der unter dem Wurzelverzeichnis befindlichen Verzeichnisse der Stationen kopiert hat, lassen sich die folgenden Schritte der Datenauswertung (Abb. 6.23 – Schritte „2“ bis „5“) unter Zuhilfenahme weiterer Batch-Dateien leicht ausführen. Der Scriptcode der dafür zuständigen Datei für die Station Eiglasdorf ist der folgenden Abb. 6.24 zu entnehmen. Die einzelnen Schritte der Aus-

wertung sind mit der gleichen Farbe hinterlegt wie in Abb. 6.23; zur leichteren Orientierung sind die verwendeten Programme zusätzlich fettgedruckt. Der Weg der einzelnen Zeitreihen lässt sich anhand der Dateinamen leicht nachvollziehen. Die Dateinamen setzen sich zusammen aus dem Anfangsbuchstaben der Station („B“, „E“, „S“, „M“, „P“) aus dem Kanal („x“, „y“ bzw. „EW“, „NS“), ggf. aus einer Kennung für den abgeschlossenen Prozess und der Extension „.dat“. Ohne diese Kennung sind die Dateien die aufbereiteten Datenreihen. Die Kennung „*Filt*“ steht für auf Stundenwerte gefilterte Reihen, „*Cal*“ für kalibriert und „*Trans*“ für transformiert. Der Übergang der Kanalkennung von „x“ und „y“ zu „EW“ und „NS“ ist ein Hinweis darauf, dass die Zeitreihen auf Nordsüd, Ostwest transformiert wurden.

```
Echo off
cd DATEN
rem *** Filtern *****
copy ..\..\BIN\FILTER.DAT
Filter.exe Ex.dat ExFilt.dat
Filter.exe Ey.dat EyFilt.dat
del FILTER.DAT
rem *** Kalibrieren des X-Kanals und Y-Kanals *****
Kalibrierung.exe ExFilt.DAT ExCal.DAT 1.65916677
Kalibrierung.exe EyFilt.DAT EyCal.DAT 1.384425422
rem *** Transformation des X-Kanals und Y-Kanals *****
Transformation.exe ExCal.DAT EyCal.DAT ENSTrans.DAT EEWTrans.DAT 138.0
rem Aendern des Datenformates: Header, Lueckenkennungen, Ende und Umbenennung in EEW, ENS
ChFormat.exe EEWTrans.DAT EEW.DAT EW
ChFormat.exe ENSTrans.DAT ENS.DAT NS
echo *** Analyse *****
rem EW-Kanal
copy ..\..\INIT\EEWproject.ew project
copy ..\..\INIT\EEW.ini
copy ..\..\INIT\EEW.ini default.ini
analyze.exe EEW
del EEW.ini
move ANALYZE.PRN ..\Ergebnisse-Analyse\Variation\ALL\AnalyzeEW.PRN
ResultCut.exe EEW.prn EEW.txt
CutLowPass.exe EEW.prn
move EEW.TXT ..\Ergebnisse-Print\Variation
move EEW.FAR ..\Ergebnisse-Analyse\Variation\ALL
move EEW.PRN ..\Ergebnisse-Analyse\Variation\ALL
move EEW.PAR ..\Ergebnisse-Analyse\Variation\ALL
move EEW.RES ..\Ergebnisse-Analyse\Variation\ALL
copy EEW.DAT ..\Ergebnisse-Analyse\Variation\ALL

rem NS-Kanal
copy ..\..\INIT\ENSproject.ns project
copy ..\..\INIT\ENS.ini
copy ..\..\INIT\ENS.ini default.ini
analyze.exe ENS
del ENS.ini
del project
del default.ini
move ANALYZE.PRN ..\Ergebnisse-Analyse\Variation\ALL\AnalyzeNS.PRN
ResultCut.exe ENS.prn ENS.txt
CutLowPass.exe ENS.prn
move ENS.TXT ..\Ergebnisse-Print\Variation
move ENS.FAR ..\Ergebnisse-Analyse\Variation\ALL
move ENS.PRN ..\Ergebnisse-Analyse\Variation\ALL
move ENS.PAR ..\Ergebnisse-Analyse\Variation\ALL
move ENS.RES ..\Ergebnisse-Analyse\Variation\ALL
copy ENS.DAT ..\Ergebnisse-Analyse\Variation\ALL

echo.
echo.
Cd..
```

Abb. 6.24: Batchdatei für die Auswertung der Datenreihen, hier am Beispiel der Station Eiglasdorf.

Nach der Gezeitenanalyse werden die Ergebnisdateien durch zwei Hilfsprogramme *ResultCut.exe* und *CutLowPass.exe* weiter bearbeitet. *ResultCut.exe* kopiert aus der Ergebnisprotokolldatei den Bereich für die Gezeitenparameter. Die Eingabeargumente sind als erstes die Quelldatei und die Zieldatei. Das Programm *CutLowPass.exe* kopiert den herausgefilterten langperiodischen Anteil der Ergebnisprotokolldatei. Das Eingabeargument ist die Quelldatei. Der Name der Zieldatei wird aus dem Namen der Quelldatei generiert. Nach dieser Extraktion werden die Ergebnisdateien in die entsprechenden Unterverzeichnisse von *\Ergebnisse-Print* kopiert.

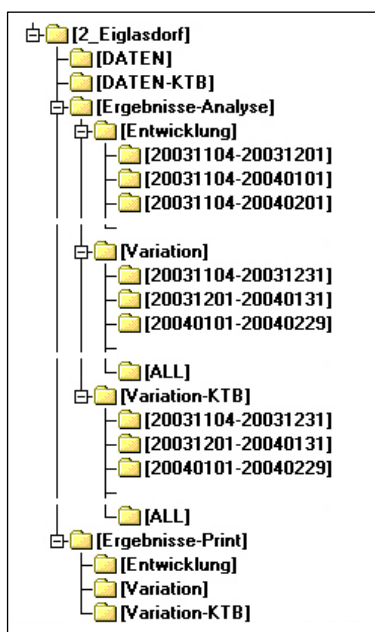


Abb. 6.25: Verzeichnisbaum der Auswertung der Datenreihen für die Station Eiglasdorf.

Die eben beschriebene Batchdatei befindet sich in dem Verzeichnis *\2_Eiglasdorf*. Das Arbeitsverzeichnis der Batch-Datei ist das Unterverzeichnis *\Daten*. Unter *\Ergebnisse-Analyse\Variation\ALL* (Abb. 6.25.) befinden sich die kopierten Ergebnisdateien und unter dem Verzeichnis *\Ergebnisse-Print\Variation\ALL* (Abb. 6.25) befinden sich die extrahierten Ergebnisdateien.

Die Verzeichnisse *\Variation* (Abb. 6.25.) allgemein sind die Speicherorte für die Variation der Gezeitenparameter in zweimonatigen Teilstücken. Diese Analyse der Teilstücke wird vorgenommen, um Veränderung durch den veränderten Porendruck

durch die Injektion nachvollziehen zu können. Die Teilstücke sind zwei Monate lang, damit die Fehler durch die Gezeitenanalyse mit *Analyze.exe* möglichst klein sind, aber auch, um noch eine genügend hohe Auflösung über die Zeit zu erhalten, und um trotzdem eine eventuelle Variation der Parameter feststellen zu können. Für die Teilanalyse wird wieder eine Batch-Datei benutzt. Der Scriptcode für die Variation ist der Abb. 6.26 zu entnehmen. Diese Batch-Datei wird mit zwei Argumenten, die eine Zeitkennung sind, aufgerufen. Ohne Eingabeargumente gibt die Batch-Datei eine Meldung über die Eingabeaufforderung aus. Die Batch-Datei legt Verzeichnisse zum Speichern der Ergebnisse und verwendeten Daten an. Der Einstiegspunkt in dem Auswertungsprozess

für die Anwendung ist das Kopieren der transformierten Datenreihen. Aus diesen werden mit Hilfe des Programms *Partcut.exe* und den Parametern der Zeitkennung die gewünschten Teilstücke aus den Datenreihen herausgeschnitten. Nach dem Ändern des Datenformates wird die Gezeitenanalyse vorgenommen und die Ergebnisse, wie schon beschrieben, in die entsprechenden Verzeichnisse kopiert.

```
Echo off

if "%1" == "" GOTO 1
if "%2" == "" GOTO 1

md Ergebnisse-Analyse\Variation\%1-%2

rem ***
copy DATEN\EEWTrans.dat
PartCut EEWTrans.dat %1 %2
CHFormat EEWTrans.dat EEW.DAT EW

rem ***
copy ..\INIT\EEW.ini
ModifyIni EEW.ini %1
copy EEW.ini default.ini
copy ..\INIT\EEWproject.ew project
analyze EEW
del EEW.ini
move ANALYZE.PRN Ergebnisse-Analyse\Variation\%1-%2\AnalyzeEW.PRN
ResultCut EEW.prn EEW%1-%2.txt
move EEW%1-%2.txt Ergebnisse-Print\Variation
move EEW.* Ergebnisse-Analyse\Variation\%1-%2
move EEWTrans.dat Ergebnisse-Analyse\Variation\%1-%2

rem ***
copy DATEN\ENSTrans.dat
PartCut ENSTrans.dat %1 %2
CHFormat ENSTrans.DAT ENS.DAT EW

rem ***
copy ..\INIT\ENS.ini
ModifyIni ENS.ini %1
copy ENS.ini default.ini
copy ..\INIT\ENSproject.ns project
analyze ENS
del ENS.ini
del default.ini
del project
move ANALYZE.PRN Ergebnisse-Analyse\Variation\%1-%2\AnalyzeNS.PRN
ResultCut ENS.prn ENS%1-%2.txt
move ENS%1-%2.txt Ergebnisse-Print\Variation
move ENS.* Ergebnisse-Analyse\Variation\%1-%2
move ENSTrans.dat Ergebnisse-Analyse\Variation\%1-%2

if not "%1" == "" GOTO 2

:1
echo TeilAnaylyse braucht zwei Zeitmarken Startzeit und Endzeit
echo (xxxxxxx; z.B. 20041123 , JahrMonatTag)
:2
```

Abb. 6.26: Batchdatei für die Teilanalyse der Variation der Gezeitenparameter der Datenreihen, hier am Beispiel der Station Eiglasdorf.

Für eine Kontrolle der Qualität der Entwicklung der Gezeitenparameter, der mit der Zeit länger werdenden Zeitreihen, gibt es die Verzeichnisse *Entwicklung* und eine ähnliche Batch-Datei, wie sie eben schon beschrieben wurde.

Zur Betrachtung der erwarteten radial zur KTB erzeugten Neigungen werden die Zeitreihen auf einem zweiten Weg der Auswertung die jeweiligen Y-Komponenten der Daten in Richtung KTB gedreht. Für diesen Teil der Auswertung gibt es zwei weitere Batch-Dateien, die ähnlich den zwei schon näher beschriebenen sind. Um die Drehung vorzunehmen, wird in einer Batch-Datei wie in Abb. 6.26 mit der Transformation begonnen. Die Verteilung der Daten Dateien erfolgt äquivalent auf die Verzeichnisse. Die Funktionsweise für die Batch-Datei der Teilanalyse erfolgt wiederum entsprechend.

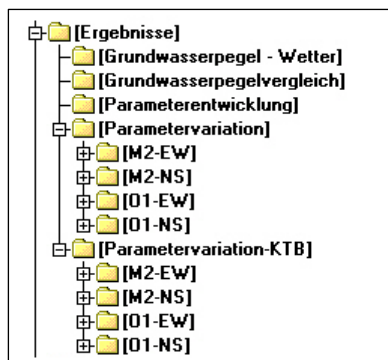


Abb. 6.27: Verzeichnisbaum von
 \Auswertung-KTB\Ergebnisse.

Das letzte hier zu behandelnde Unterverzeichnis ist \Ergebnisse (Abb. 6.27). Hier sind, wie schon der Name sagt, die Ergebnisse der Analysen und Aufbereitungen zu finden. Hauptsächlich sind hier Graphiken und Tabellen zu finden. Die Graphiken sind mit Hilfe von *Grapher 2.0* erstellt wurden. Auf die Unterverzeichnisse soll nicht eingegangen werden, da die Namen für sich selbst sprechen. Im Anhang V ist eine Aufstellung

der zur Auswertung der Daten notwendigen Batch-Dateien und Programme zu finden.

6.4 Aufbereitung der Wetter- und Grundwasserdaten

Nachdem die Daten der Grundwassermessstationen von den Divern heruntergeladen und Luftdruck kompensiert worden sind (Kapitel 5.2), müssen diese weiter aufbereitet werden. Dies ist notwendig, um die Messungen mit den Datenreihen der Neigungsmesser zu vergleichen. Als Erstes werden die Textdateien von *LDM* (siehe Kapitel 5.2) in das ETERNA-Format mit Hilfe des Programms *MonConvert.exe* konvertiert. Das Programm wurde einzig und allein für diese Aufgabe entwickelt und kann auch nur bei Dateien angewendet werden, welche von *LDM* herausgegeben wurden. Im folgenden Schritt wurden die Datenreihen mit *PreAnalyse.exe* gesichtet und nötigenfalls editiert, was im eigentlichen Sinn nicht notwendig war. Die Datenreihen hatten in der Regel eine Samplerate von 5 Minuten. Nur in der Zeit vom 22.12.2004 bis zum 30.01.2005

lag die Samplerate bei 1 Minute. Für diesen Zeitraum wurden die Datenreihen erst einmal auf 5 Minutendaten gefiltert. Dies geschah mit dem schon unter 6.2.2 beschriebenen Programm *Filter.exe*. Dabei wurde ein Filter verwendet, der von Wenzel (1996) entwickelt worden war. Die Charakteristik dieses Filters ist in Abbildung 6.28 dargestellt. Die nun vorliegenden Teilstücke in 5 Minutendaten wurden wiederum mit *EternaAdd.exe* (Kapitel 6.2.1.3) zusammengefügt. Nach einer nochmaligen Sichtung der Datenreihen mit *PreAnalyse.exe* wurden sie dann auf Stundenwerte gefiltert. Der entsprechende Filter (Abb. 6.29) geht wiederum auf Wenzel (1996) zurück. Nun standen die Datenreihen in dem gleichen Format wie die Daten der Neigungsmesser zur Verfügung und konnten verglichen werden.

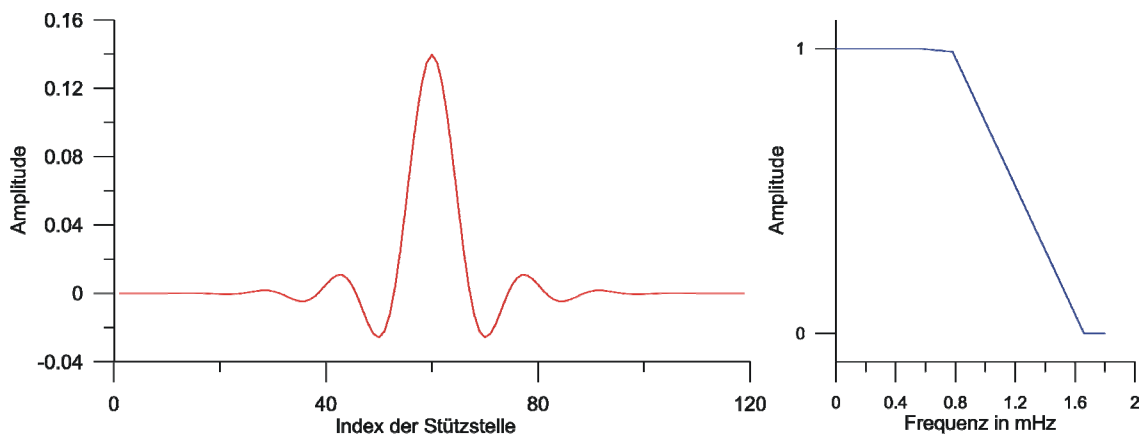


Abb. 6.28: Filteroperator für Filterung von 1 Minutendaten auf 5 Minutendaten mit 119 Stützstellen (links) und Frequenzcharakteristik (rechts - idealisiert) (nach Wenzel, 1996).

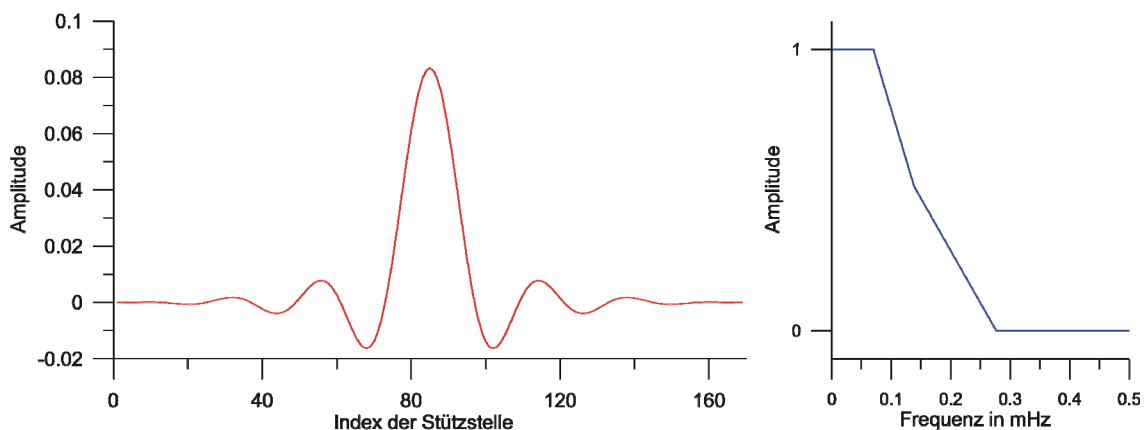


Abb. 6.29: Filteroperator für Filterung von 5 Minutendaten auf Stundendaten mit 169 Stützstellen (links) und Frequenzcharakteristik (rechts - idealisiert). (nach Wenzel, 1996).

Im Vergleich dieser beiden Filteroperatoren mit dem 10 Sekunden – 1 Stunden Filter (Kapitel 6.2.2) sind sie mit Längen von 2 bzw. 14 Stunden kürzer. Der hauptsächliche Grund dafür liegt in der Übertragungsfunktion. Die Flanke des Übertragungsbereiches ist im 10 Sekunden – 1 Stunden Filter steiler.

Die Aufbereitung der Wetterdaten war einfacher gelagert als die der Neigungsmesser- und Grundwasserdaten. Nach Einrichtung der Wetterstation am 01.06.2004 betrug bis zum 3.11.2004 die Samplerate eine Stunde. Danach wurde sie auf eine halbe Stunde gesenkt, um eine bessere Auflösung der Wetterparameter über die Zeit zu erhalten. Die Erniedrigung der Samplerate auf eine halbe Stunde ist aber nicht sinnvoll, da die Verbindung zu den Neigungsdaten erst dann erfolgt wenn diese in Stundenwerten vorliegen. Um nun die entsprechenden Werte für Temperatur, Niederschlag und Luftdruck einfach aus der von *Weatherlink* (Kapitel 5.3) generierten Textdatei extrahieren zu können, war es angebracht, das Programm *Reduse.exe* zu entwickeln. Das Programm unterscheidet selbst, welche Samplerate gerade vorliegt. Ist diese eine Stunde, so werden die gewünschten Werte einfach im ETERNA-Format in Dateien geschrieben. Ist die Samplerate aber eine halbe Stunde, so nimmt das Programm für die Temperatur und den Luftdruck eine Mittelung der Werte der Halbstunden- und Stundenwerte vor. Für den Niederschlag werden die Werte der Halbstunden- zu den Stundenwerten addiert. Nun liegen die Zeitreihen der Wetterdaten im gleichen Format vor wie die Neigungs- und Grundwasserdaten und stehen einer weiteren Verwendung zur Verfügung.

7 Verlauf der Injektion

Der geplante Verlauf der Injektion (Kapitel 1.3) konnte nach anfänglichen Tests nicht eingehalten werden. So wurde mit der maximalen Pumpleistung gearbeitet, um die erwarteten Effekte registrieren zu können. Abb. 7.1 (oben) zeigt den Verlauf des Druckes am Kopf der Vorbohrung. Während der Injektionszeit wurden kontinuierlich ca. 200 l/min Wasser verpresst. Vom 6.-9. Oktober 2004 wurde mit einer zweiten Pumpe der BGR injiziert, dabei lag der maximal erreichte Druck bei 12.57 MPa. Neben der Druckkurve ist eine angepasste idealisierte Kurve (Abb. 7.1 oben) zusehen. Das Abfallen des Druckes nach Beginn der Injektion ist nach Kämpel (2005)¹ auf die Abkühlung des Gebirges aufgrund der Injektion von kaltem Wasser, und somit thermomechanische Kontraktion und Öffnung der Wegsamkeiten (lokal bis regional, um die Injektionsstelle herum) zurückzuführen. Bei sehr langer Injektionsdauer wird der Effekt zunehmend durch Druckaufbau aufgrund der insgesamt begrenzten Aufnahmekapazität des Gebirges überlagert. In der Mitte der Abb. 7.1 ist die Menge des induzierten Fluids gezeigt. Insgesamt wurden während der 320 Injektionstage 85000 Liter Wasser verpresst. Vor der Injektion fand ein Pumpexperiment in der Hauptbohrung statt, wobei der Wasserstand auf -50 m abgesenkt worden war. Der Wasserstand (Abb. 7.1 unten) stieg nach Ende des Experiments kontinuierlich an. Ab Injektionsbeginn stieg der Wasserstand schneller an, bis am 23. September 2004 die Hauptbohrung artesisch wurde. Das Fluid konnte ungehindert ausfließen. Am 27. Dezember 2004 wurde die Hauptbohrung für 7 Tage verschlossen, und der Druck stieg bis auf ca. 0.8 MPa an. Dies entspricht einer Wassersäule von ca. 80 m. Unter Zuhilfenahme der Messwerte wurde mit *FEFlow* eine Modellkurve angepasst (Gräsle, 2005).

¹ Persönliche Mitteilung

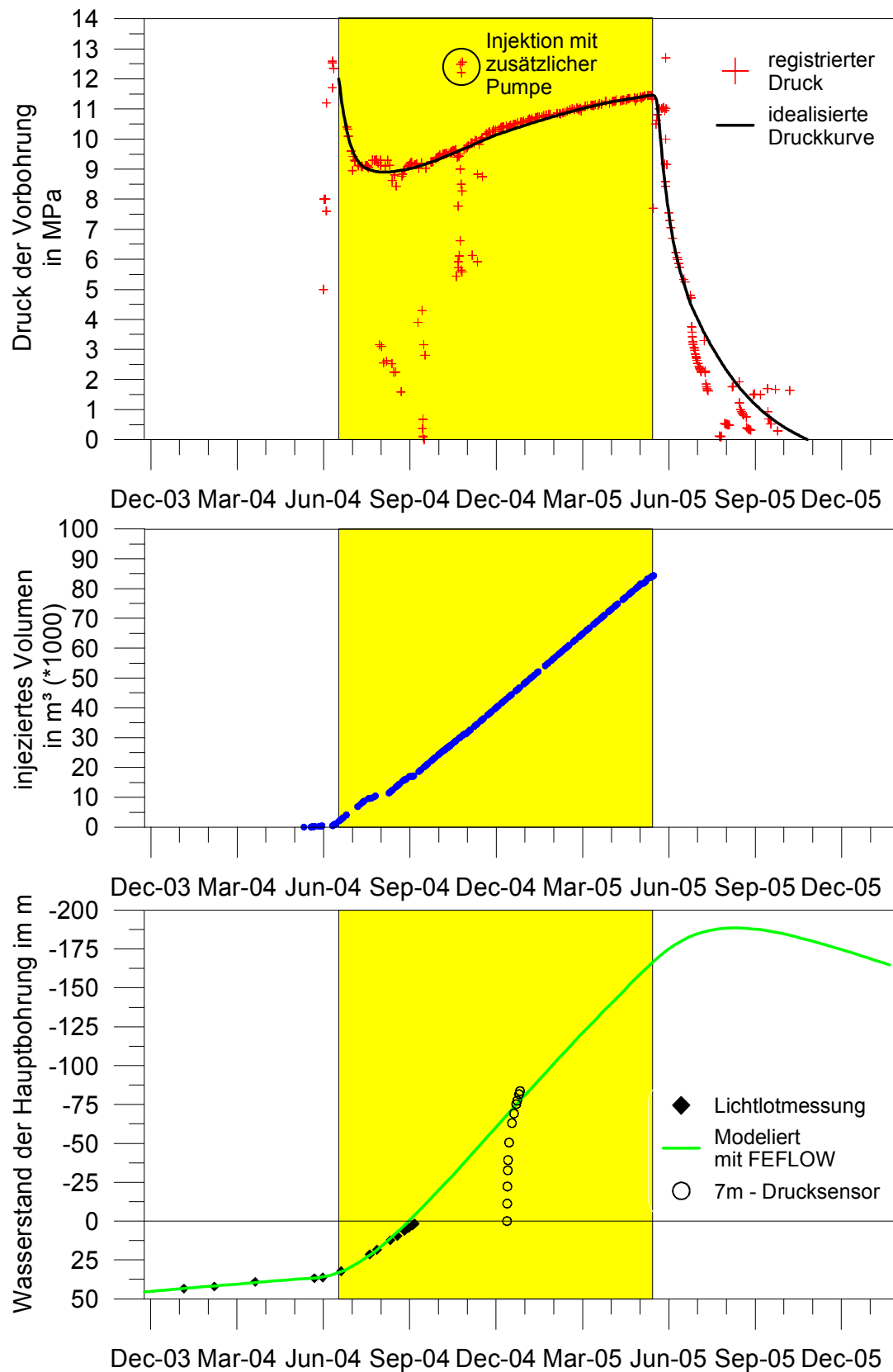


Abb. 7.1: Verlauf der Injektion (oben – Druckverlauf an der Vorbohrung, mitte – Menge des injizierten Volumens, unten – Wasserstand an der Hauptbohrung (nach Gräsle, 2005), gelb Unterlegter Bereich ist der Injektionszeitraum).

8 Datengrundlage und -analyse

8.1 Datengrundlage

Im folgenden Kapitel sollen die mit dem unter Kapitel 6 beschriebenen Verfahren aufbereiteten und analysierten Zeitreihen diskutiert werden. Die vorhandenen Rohdaten der Neigungsmesser und der Grundwassersensoren sind tabellarisch im Anhang V aufgelistet.

Die hier verwendeten Daten der Neigungsmesser beginnen im Allgemeinen Anfang November 2003 und reichen bis Ende Dezember 2005. Die Registrierung der Neigungsmesserdaten an der Station Mittelberg hat einen Monat später begonnen und in Stockau hat sie Ende Juli 2005 aufgehört. Die Grundwassersensoren an den jeweiligen Stationen wurden Anfang Mai 2004 installiert und die meteorologische Station an der KTB hat ihre Arbeit Anfang Juni 2005 aufgenommen. Die Registrierzeiträume sind in Abbildung 8.1 dargestellt. Zum Vergleich ist die Dauer der Injektion in der Graphik hinzugefügt.

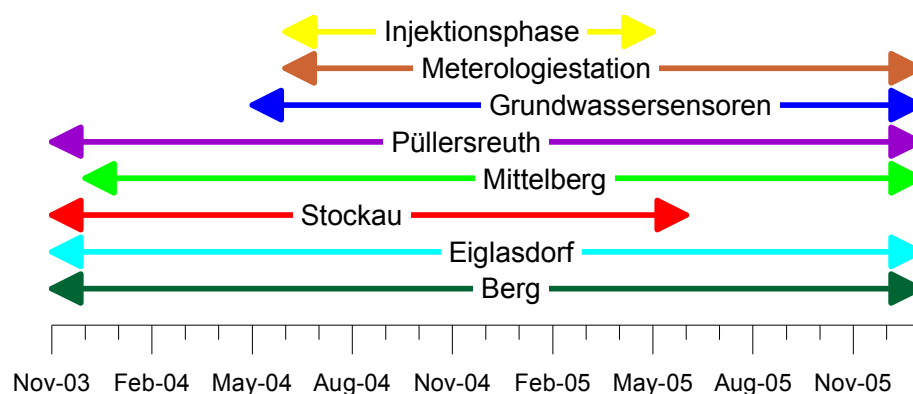


Abb. 8.1: Zeiträume der Registrierungen der Messgeräte und Dauer der Injektion.

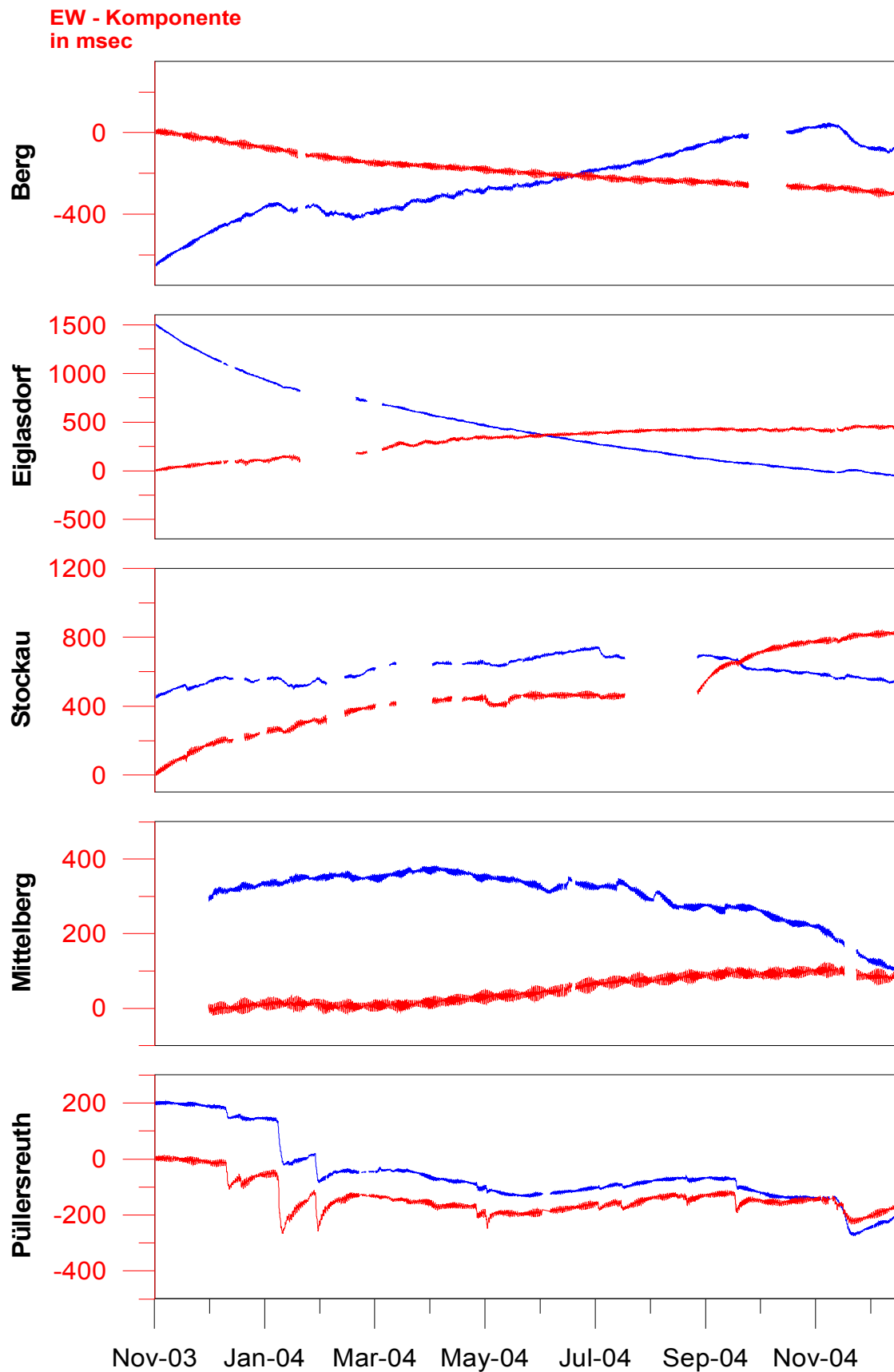
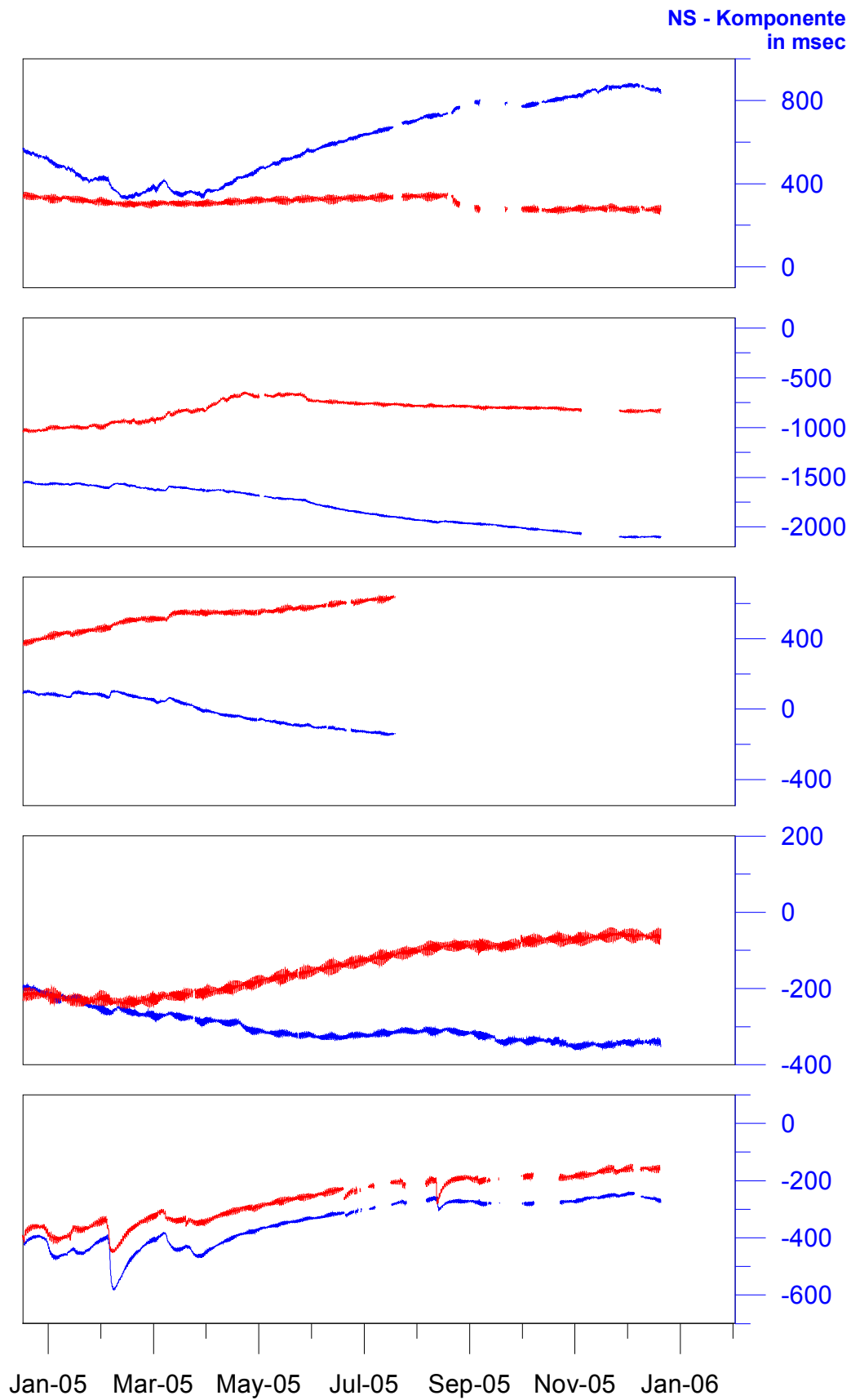


Abb. 8.2: Registrierungen der Neigungsmesser, kalibriert, transformiert auf NS-EW und gefiltert auf Stundenwerte.



In Tabelle 8.1 ist eine Aufstellung der verwendeten Daten zu finden. Ebenso sind die in den Daten enthaltenen Lücken verzeichnet. Die Station Stockau hat mit 16.3% fehlenden Daten der Zeitreihe die größten Ausfälle, Mittelberg hingegen mit 1.7% die geringsten. Die während der 788 Registriertage entstandenen Lücken sind meist auf technische Probleme zurückzuführen.

Tab. 8.1: Aufstellung der Länge der verwendeten Daten und der vorhandenen Lücken.

Berg		Eiglasdorf		Stockau		Mittelberg		Püllersreuth	
Länge der vorhandenen Zeitreihen:									
788 d		788 d		633 d		759 d		788 d	
davon Lücken in Zeitreihen:									
Beginn	Dauer in h	Beginn	Dauer in h	Beginn	Dauer in h	Beginn	Dauer in h	Beginn	Dauer in h
18.11.03	6	24.11.03	6	01.12.03	9	17.06.04	7	07.02.04	16
22.01.04	107	10.12.03	27	17.12.03	153	18.06.04	17	25.02.04	37
23.05.04	7	14.12.03	95	31.12.03	100	23.06.04	49	29.02.04	36
30.09.04	506	23.01.04	747	07.02.04	17	19.11.04	12	05.03.04	17
03.01.05	14	29.02.04	204	08.02.04	223	22.11.04	164	23.05.04	8
28.07.05	126	17.11.04	26	05.03.04	189	15.01.05	9	07.06.04	103
29.08.05	55	09.04.05	15	17.03.04	485	02.04.05	23	09.11.04	29
05.09.05	208	18.04.05	12	17.04.04	144	02.06.05	18	14.11.04	50
17.09.05	351	11.05.05	67	05.05.04	11	22.12.05	11	09.12.04	11
03.10.05	201	15.11.05	530	18.05.04	13			13.03.05	8
21.10.05	41	19.12.05	12	26.05.04	18			29.06.05	10
30.11.05	14			23.07.04	971			05.07.05	14
19.12.05	23			15.11.04	11			07.07.05	38
22.12.05	16			16.11.04	14			10.07.05	108
				09.05.05	14			17.07.05	228
				19.06.05	24			05.08.05	275
				01.07.05	65			21.09.05	19
				28.07.05	13			23.09.05	115
								28.09.05	312
								12.10.05	17
								18.10.05	365
								16.12.05	96
Gesamtgröße der Lücken:									
1675		1741		2474		310		1912	
69 d 19 h		72 d 13h		103 d 2 h		12 d 22 h		79 d 16 h	
Anteil der Lücken der gesamten Zeitreihe:									
8.8%		9.1%		16.3%		1.7%		10.0%	

In Abbildung 8.2 sind die Stundendaten der Neigungsmesser zu sehen. Nachdem die Zeitreihen der Kanäle der 5 Stationen von Sprüngen und Spikes befreit wurden, wurden sie von 10 Sekunden- auf Stundenwerte gefiltert, kalibriert und auf Nordsüd-

Ostwest transformiert. Die Y-Achsen geben die wahren Neigungen der Erdoberfläche an den Messstationen relativ zur Erdoberfläche zu Beginn der Registrierung in Millisekunden (msec) an. Die EW und NS-Achsen der Diagramme jeder Station sind gleich skaliert, die einzelnen Stationen sind dagegen unterschiedlich skaliert. Die zweiwöchige Schwebung in der Gezeitenamplituden, welche deutlich in Berg und Mittelberg zu erkennen ist, beträgt ca. 20 msec (~ 100 nrad).

Die Zeitreihen (Abb. 8.2) beinhalten Driften, welche sich durch lineare bzw. logarithmische Funktion approximieren lassen. Woher dieser Effekt kommt, lässt sich noch nicht mit Sicherheit sagen. In den Daten der Station Püllersreuth sind sägezahnartige Effekte zu sehen, welche meteorologischen Effekten bzw. Änderungen im Grundwasser zugeschrieben werden können (Weise, 1992).

Im September 2005 musste das unter Kapitel 5.1.7 beschriebene Registriersystem umgestellt werden. Das Funknetzwerk (WLAN) und die EDL wurden abgebaut und durch PCs mit DCF-Uhr ersetzt. Bedingt durch diesen Umbau entstanden durch Ausfälle dieser PCs Lücken.

Zusätzlich zu den Neigungen wurden an der kleinen Wetterstation an der KTB weitere Parameter registriert wie die Temperatur, der Niederschlag, der Luftdruck sowie Windrichtung und -stärke (Kapitel 5.3). Der Grundwasserdruck (-pegel) (Kapitel 5.2) wurde direkt an den Stationen gemessen. Die in diesem Zusammenhang gewonnenen Daten zeigt Abbildung 8.3. Im November 2005 ist durch Ausfall der Wetterstation eine 2-wöchige Lücke in den Daten entstanden.

Durch den Vergleich des Niederschlages mit den Grundwasserpegeln der fünf Stationen lassen sich viele Variationen der Wasserstände einem Niederschlagsereignis zuordnen. Nimmt man die Temperatur mit ihrer Jahreswelle hinzu, so können Gemeinsamkeiten in den Graphen erkannt werden. Allgemein reagieren die Wasserstände unterschiedlich auf die Niederschlagsereignisse, wobei der Pegel in Püllersreuth am stärksten, in Berg geringer und an den anderen Stationen kaum reagierte.

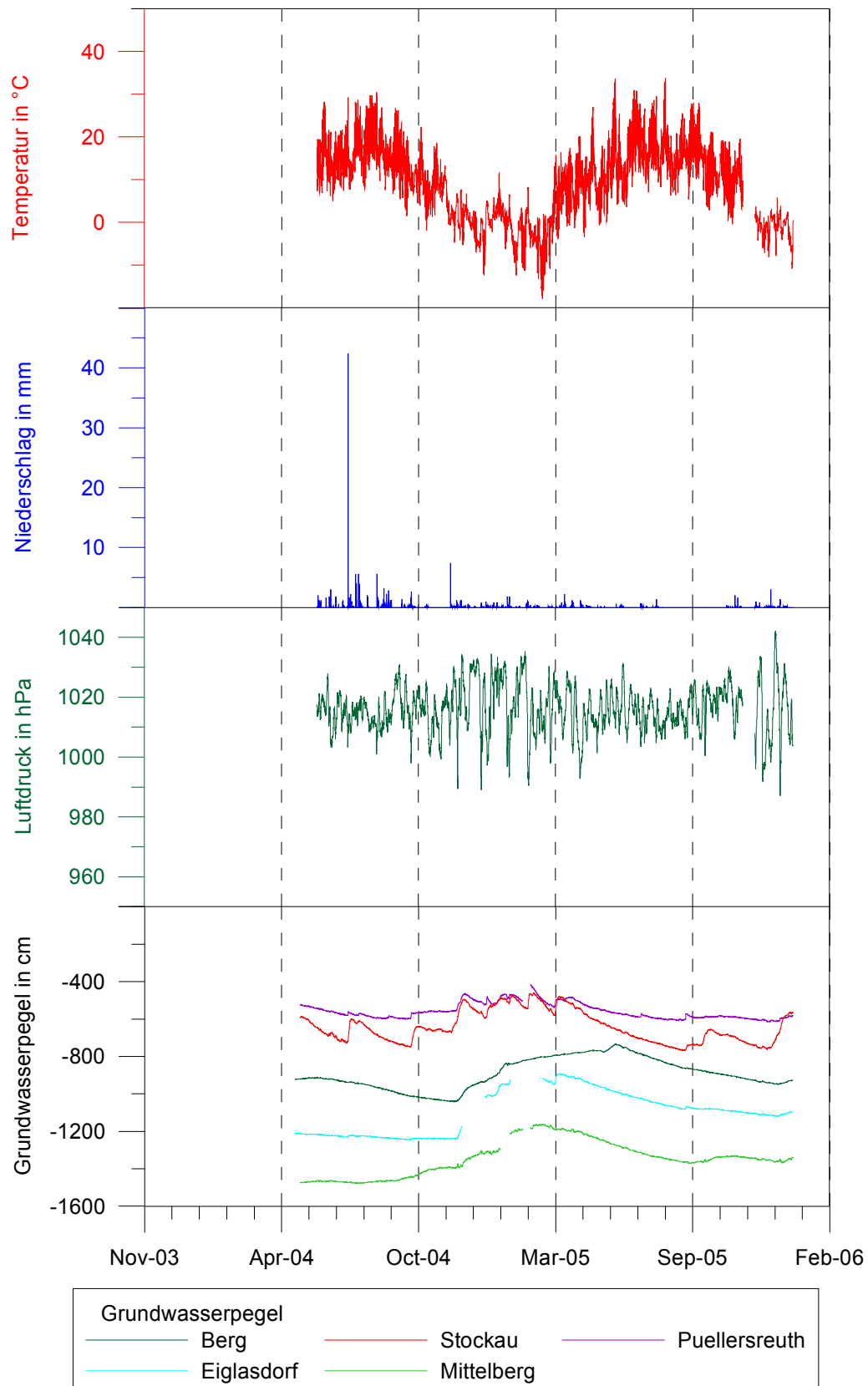


Abb. 8.3: Vergleich Temperatur, Niederschlag, Luftdruck und Grundwasserpegel der Messstationen.

8.2 Analyse der Gezeitenparameter

Die aufbereiteten Zeitreihen (Abb. 8.2) standen nun für die Gezeitenanalyse (Kapitel 6.2.5) zur Verfügung. In den ersten Analysen wurden die Daten in der Weise analysiert, dass die Zeitreihen stetig einen Monat länger wurden. In Abbildung 8.4 sind die resultierenden Gezeitenparameter zu finden. Der Parameter ist der Quotient aus der beobachteten Amplitude der jeweiligen Welle und der theoretischen Amplitude. Bei Neigungsmessungen liegt er für eine Standarderde bei 0.7. Bei der Erstellung der Diagramme wurden nur die Parameter der größten ganztägigen (O1) und der halbtägigen (M2) Welle berücksichtigt. Bei der Analyse werden die theoretischen Gezeitenwellen an das beobachtete Signal über die Methode der kleinsten Quadrate angepasst, unter der Variation des Gezeitenparameters und der Phase. Für die Auswertung der unterschiedlichen Gezeitenwellen gibt es in Abhängigkeit von der zu untersuchenden Welle (Anhang III) Untergrenzen der Länge der zu analysierenden Zeitreihen.

Insgesamt lässt sich der Abbildung 8.4 die Qualität der Zeitreihen entnehmen. Wie zu erwarten ist, werden die Fehler der Gezeitenanalysen mit länger werdenden Zeitreihen kleiner. Bedingt durch das Analyseverfahren und das verwendete Gezeitenmodell ist es durchaus möglich, dass bei Zeitreihen, die länger als ein Jahr sind, die Fehler der Gezeitenparameter wieder größer werden können (Weise, 2006)¹. Dies ist in der ganztägigen Welle der EW-Komponente der Stationen Eiglasdorf und Stockau der Fall.

In den Analysen der Stationen Berg und Stockau der NS-Komponente der O1-Welle zeigt sich eine vergleichsweise große Streuung der Gezeitenparameter, deren Ursache unbekannt ist. Aufeinanderfolgende Analysen liegen aber in jedem Fall in den jeweiligen Fehlergrenzen der vorangegangenen und nachfolgenden Intervalle. Die Station Mittelberg bildet hier unter allen ein ausgezeichnetes Beispiel. Die Gezeitenparameter sind über die Zeit sehr stabil und haben sehr kleine Fehler.

¹ Persönliche Mitteilung

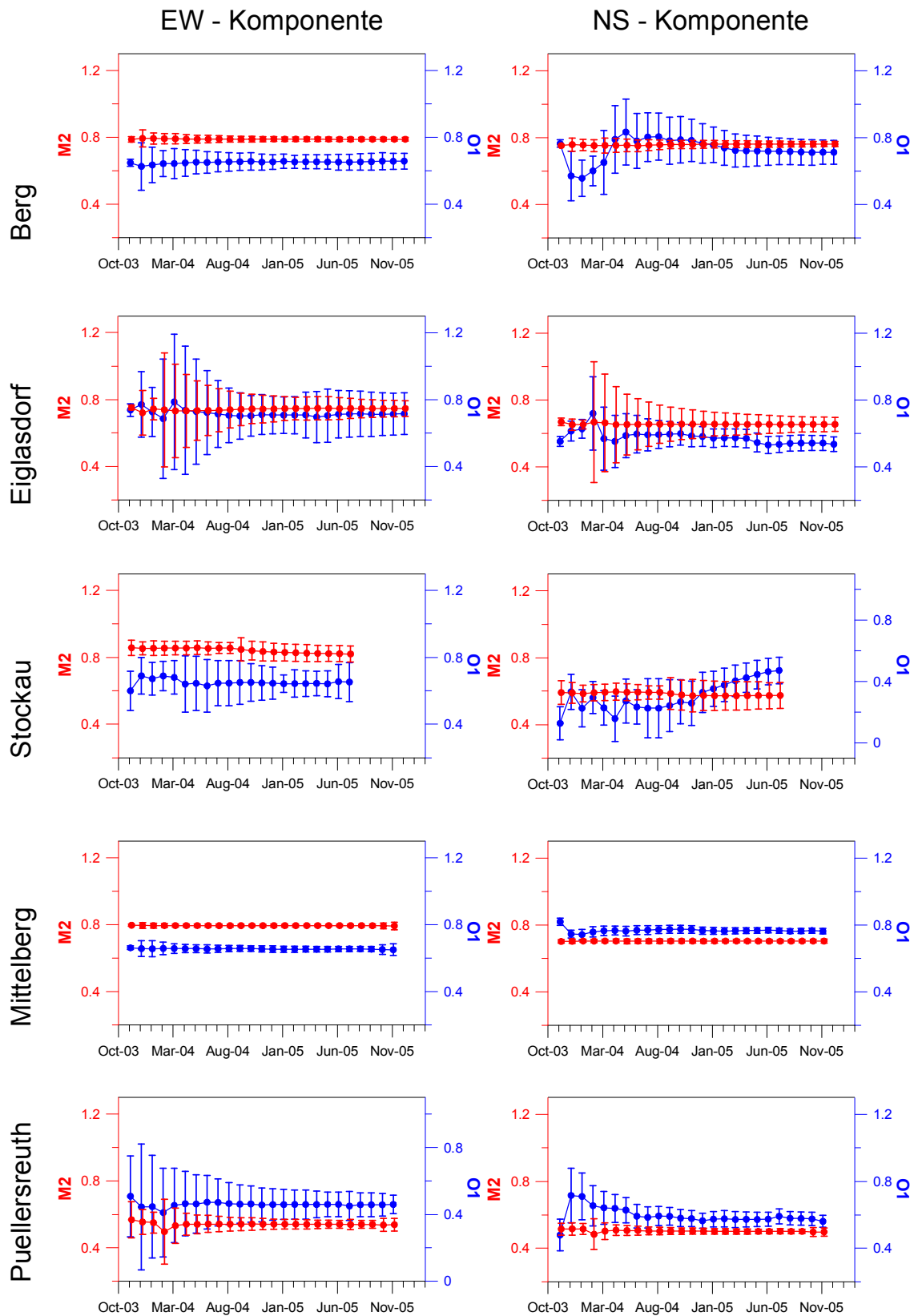


Abb. 8.4: Entwicklung der Gezeitenparameter der Gezeitenwellen O1 und M2 für länger werdende Zeitreihen.

Ein weiterer Ansatz um Aussagen im Rahmen des Injektionsprojektes zu gewinnen, ist die Analyse von Teilstücken von zwei Monaten Länge, welche sich jeweils um einen Monat überlappen (Abb. 8.5, oben). Der gelb hinterlegte Bereich markiert den Zeitraum der Injektion. In den Analysen lässt sich kein Trend erkennen. Erwarten lässt sich, dass sich mit der Injektion der Gezeitenparameter ändert. Bedingt durch das eingebrachte Volumen des Wassers und des gestiegenen Porendruckes während der Injektion sollte das betrachtete System starrer reagieren. Für den Gezeitenparameter bedeutet dies eine Verkleinerung (siehe Kapitel 9, Diskussion S. 131).

Alle Parameter der Stationen liegen innerhalb der Fehler, so dass aus den Diagrammen keine signifikante Aussage getroffen werden kann. Da die Qualität der ermittelten Gezeitenparameter nicht nur von den bestimmten Fehlern gegeben wird, sondern auch sehr stark von der Länge der zur Bestimmung verwendeten Zeitreihen abhängt, ist es notwendig, weitere Betrachtungen in dieser Hinsicht anzustellen. Zu diesem Zweck wurden die ermittelten Gezeitenparameter (γ) für jeden Zeitpunkt über die Formel (8.1) gemittelt, wobei die einzelnen γ -Faktoren über die Fehler gewichtet wurde. Die Variable γ_R bezeichnet das Resultat und n ist die Anzahl der zu mittelnden Werte. Der Gezeitenparameter der entsprechenden Welle ist γ_w und der dazugehörige Fehler ist γ_{wf} .

$$\gamma_R = \frac{\sum_0^n \left(\gamma_w \cdot \frac{1}{\gamma_{wf}} \right)}{n \cdot \sum_n \frac{1}{\gamma_{wf}}} \quad (8.1)$$

An die so bestimmten Graphen der gemittelten Parameter von O1 und M2 wurden mittels linearer Regression Geraden für die Zeiträume vor, während und nach der Injektionsphase angepasst. Das Ergebnis der Mittelung ist in Abbildung 8.5 (unten) zu sehen. Die Anstiege der Geraden sind in entsprechenden Farben dargestellt. In der NS-Komponente ändert sich mit Beginn der Injektion der Anstieg der angepassten Geraden von negativ zu positiv. Sind dies Hinweise, dass die NS-Richtung sich gegenüber der EW-Richtung auszeichnet (siehe Kapitel 9)?

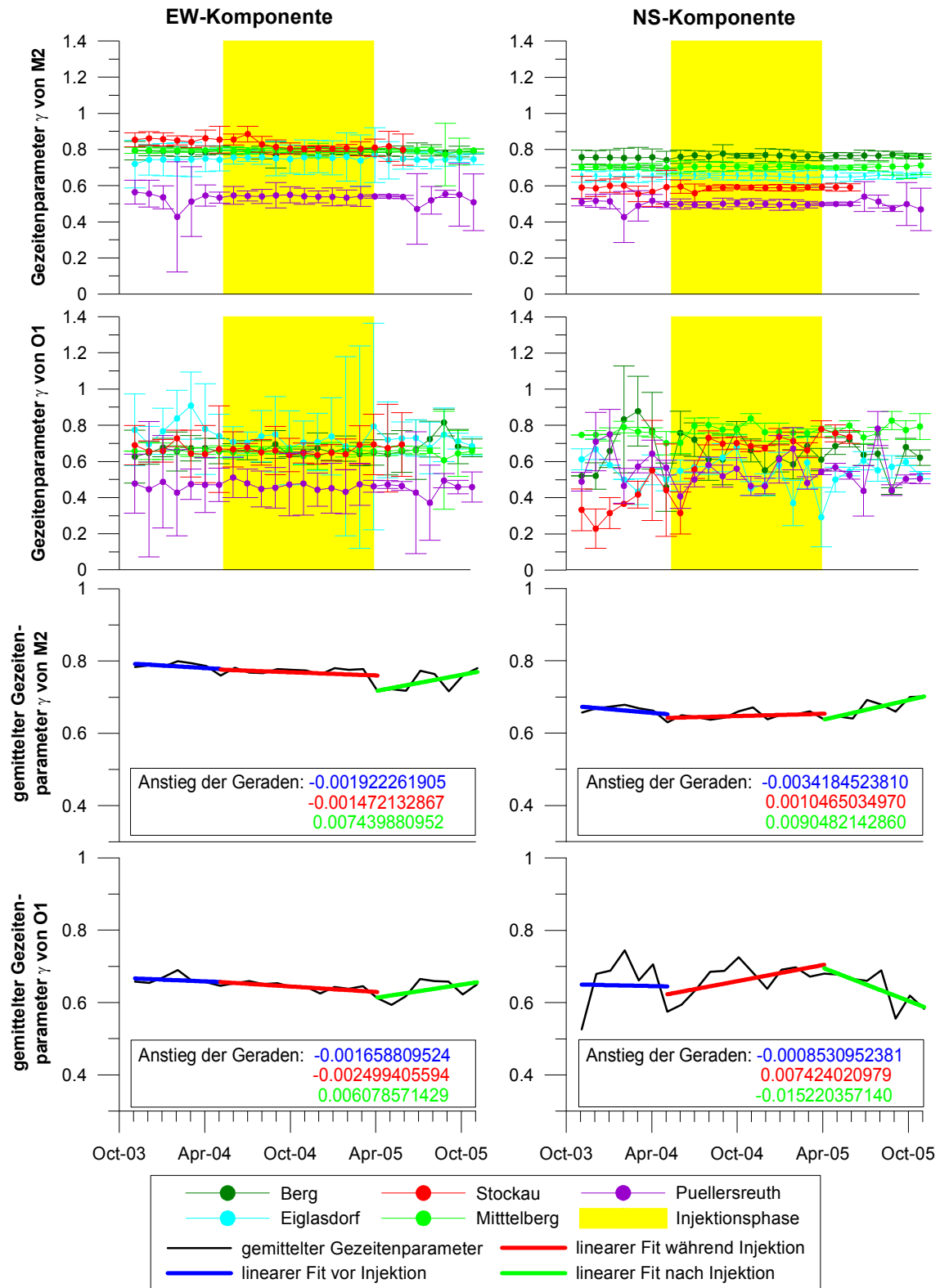


Abb. 8.5: Variation der Gezeitenparameter durch zwei Monatsanalysen (4 obere Diagramme) und gemittelter Gezeitenparameter mit angepassten Geraden für die entsprechenden Zeiträume vor, während und nach der Injektion der zwei Monats-Teilanalysen (4 untere Diagramme).

8.3 Driften der Zeitreihen

Bei der Gezeitenanalyse werden durch den Pertsev-Filter (Kapitel 6.2.5) die langperiodischen Anteile der Daten von den Gezeitenwellen getrennt. Diese Drift lässt sich mit den Grundwasserpegel-Änderungen korrelieren (Abb. 8.6).

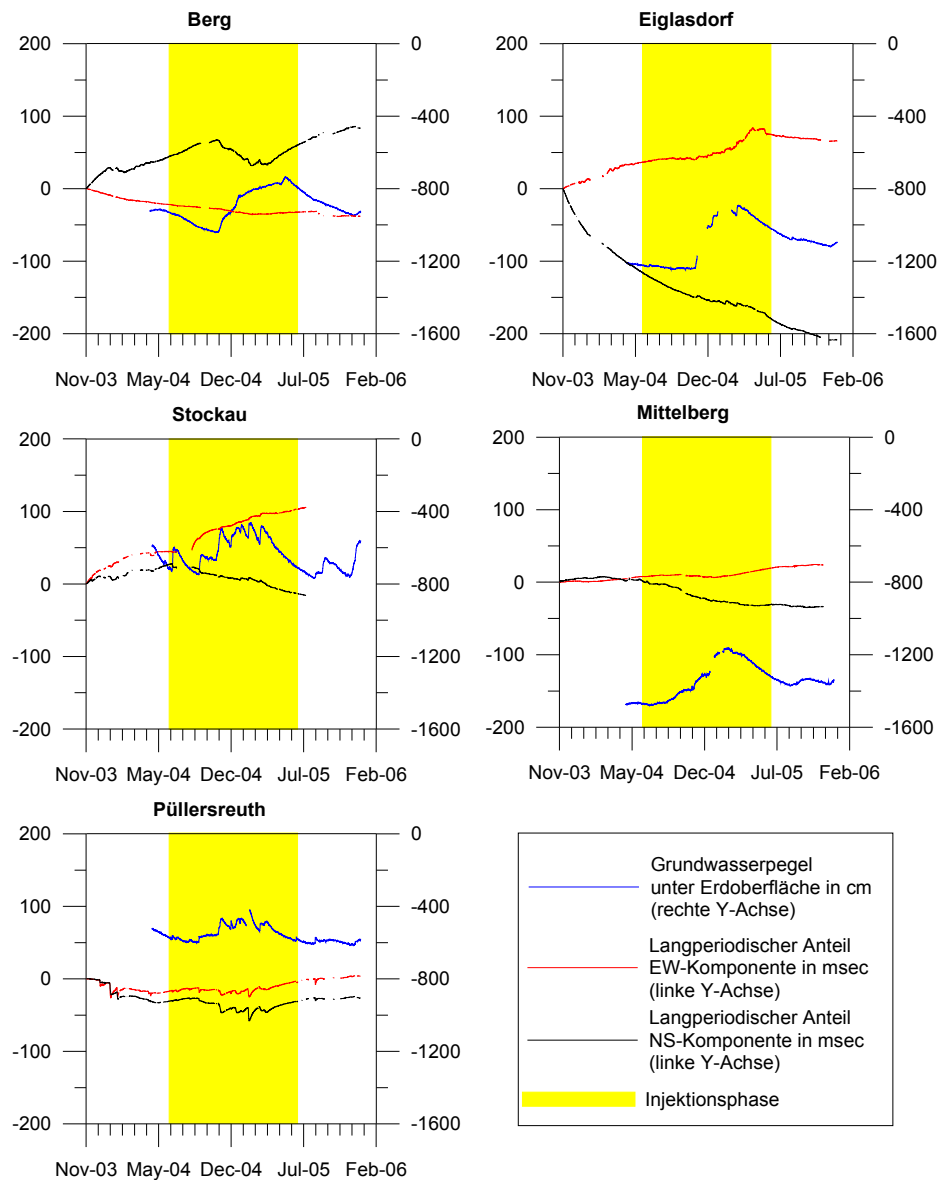


Abb. 8.6: Vergleich der langperiodischen Anteile der Komponenten (Drift) mit den Grundwasserständen.

In Püllersreuth ist die Korrelation zwischen den Porendruckvariationen, welche durch einzelne Niederschlagsereignisse ausgelöst werden und dem Neigungssignal am

größten. Ein Grund dafür ist, dass diese Station die flachste Instrumentenbohrung (Unterkante Instrument 24.5 m) aufweist und weil die Zeitreihen von Driften überlagert sind, wobei in den Stationen Berg, Eiglasdorf und Mittelberg keine signifikanten kurzperiodischen Auswirkungen von Porendruckänderungen zu erkennen sind. Zur Reduktion der Driften wurden an die Zeitreihen der Stundendaten lineare bzw. logarithmische Funktionen angepasst. Beispielhaft soll dies für die NS-Komponente der Station Berg (Abb. 8.7) gezeigt werden.

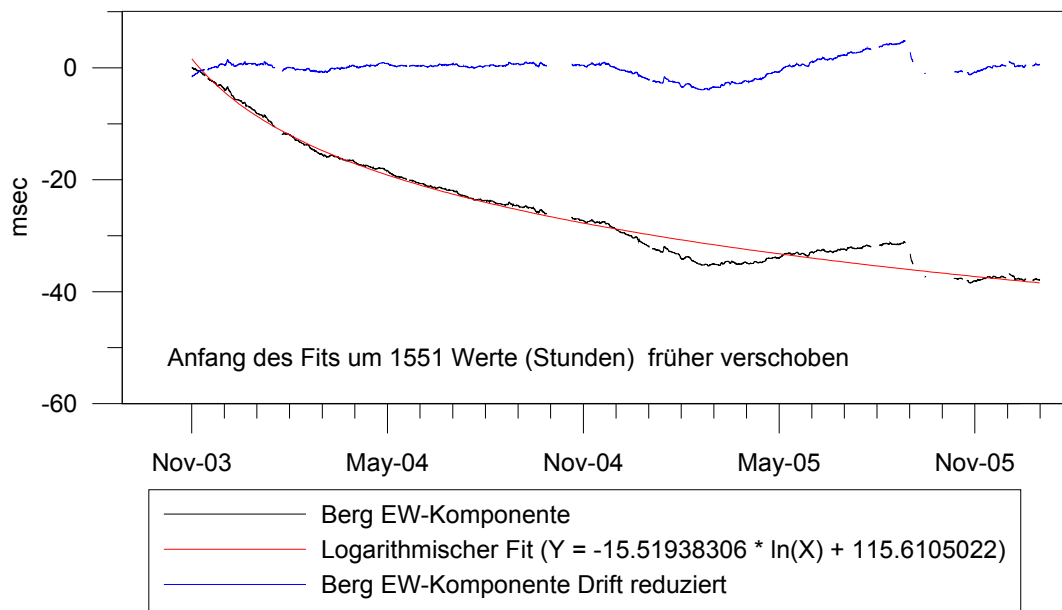


Abb. 8.7: Reduktion der Drift der EW-Komponente der Station Berg.

Nach der Reduktion der Driften (Abb. 8.8) ist nun in allen Stationen der Einfluss des Porendruckes zu erkennen. In der Station Stockau sind auch die kurzperiodischen Änderungen erkennbar. Bei der Anpassung der EW-Komponente der Daten des zweiten Instrumentes (siehe Kapitel 5.1.4) gab es Probleme für den Zeitraum der ersten sechs Wochen. Hier ist ein negativer Ausschlag in den Daten zu erkennen (Abb. 8.8) (mehr dazu in Kapitel 9). Die erwarteten Neigungssignale, ausgelöst durch die Injektion sind aber nicht klar erkennbar, da sie, wenn vorhanden, noch von Grundwassereffekten überlagert sind.

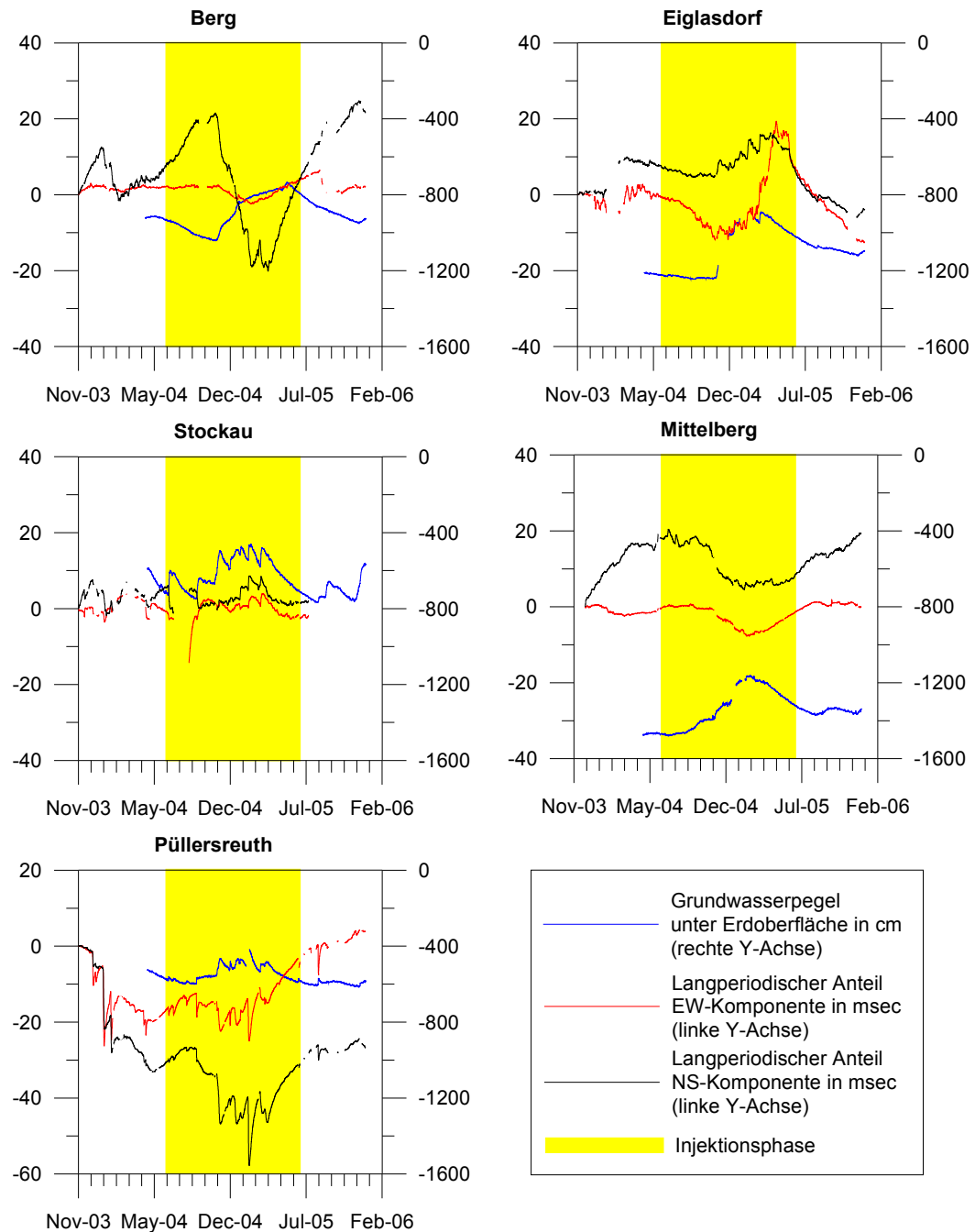


Abb. 8.8: Vergleich der langperiodischen Anteile der Komponenten (Drift reduziert) mit den Grundwasserständen.

Der Neigungseffekt, der durch die Injektion ausgelöst wird (Abb. 1.2), erscheint in den radialen Komponenten, der auf die KTB transformierten Kanäle am deutlichsten (Abb. 8.9, oben). Weil er von der Größenordnung her von den alles überlagernden Driften unterdrückt wird, sind die Zeitreihen in einem weiteren Schritt reduziert (wie in Abb. 8.7 exemplarisch gezeigt) worden (Abb. 8.9, unten).

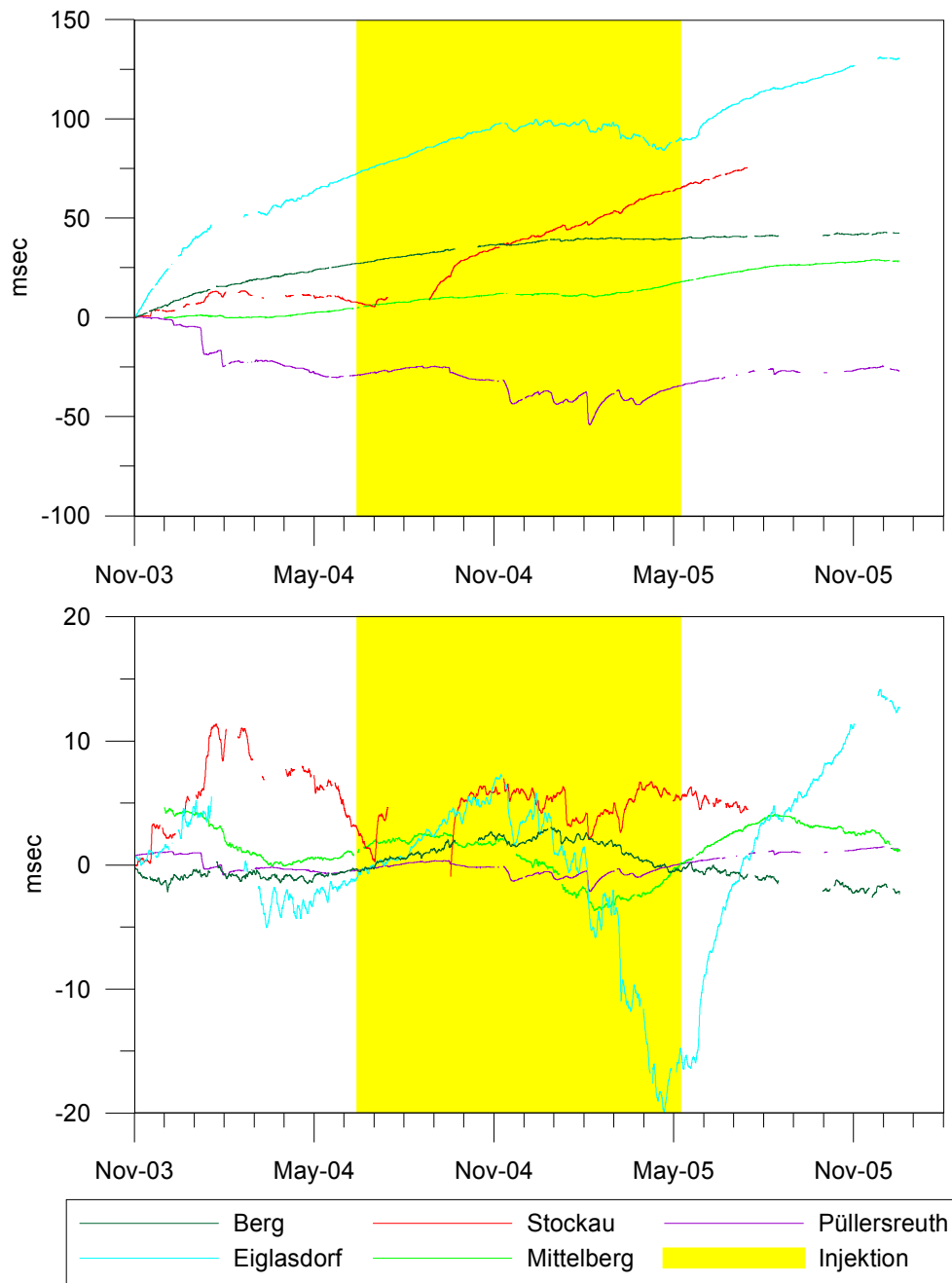


Abb. 8.9: Radiale Komponenten der zur KTB transformierten Kanäle (oben), und Drift reduzierte Daten (unten).

Infolge des starken Einflusses des Porendrucks (Grundwasserpegel) auf die Neigungen wurde versucht, diesen zu reduzieren. Hierzu gibt es mehrere Möglichkeiten. Da hier nur der langperiodische Anteil von Bedeutung ist, wird davon ausgegangen, dass das Grundwasser und die Neigungen ohne Phasenverschiebung auftreten. Daher lässt sich das Grundwasser gw mittels eines Faktors b korrelieren und von den Zeitreihen der Neigungen α abziehen (Gl. 8.2).

$$\alpha_R = \alpha - (gw \cdot b) \quad (8.2)$$

Um b zu bestimmen kann man verschiedene Werte ausprobieren und den Wert verwenden, der den geringsten Einfluss der beiden Zeitreihen auf die resultierenden Neigungen α_R aufweist. Eine andere Möglichkeit ist, in einem Diagramm die Neigung gegenüber dem Grundwasser zu plotten und eine lineare Approximation vorzunehmen (Abb. 8.10, 8.11, 8.12 oben). Der Anstieg der Geraden ist der benötigte Faktor b . Da die Stationen Berg, Eiglasdorf und Mittelberg in diesem Rahmen die besten Daten geliefert haben, wurde sich nur auf diese Stationen beschränkt. In den Abbildungen 8.10, 8.11, 8.12 (unten) sind die radialen Neigungen, das Grundwasser und die reduzierten Neigungen abgebildet. In Mittelberg ist die Reduktion am deutlichsten sichtbar, in den anderen beiden Stationen dagegen sehr gering. Dies liegt an der jeweils sehr geringen Korrelation der Neigung und des Grundwassers. Bei Versuchen mit anderen Faktoren b hat sich gezeigt, dass die ermittelten Faktoren gut passen; nur in Mittelberg war dieser etwas zu klein und wurde nachkorrigiert (Abb. 8.12, oben).

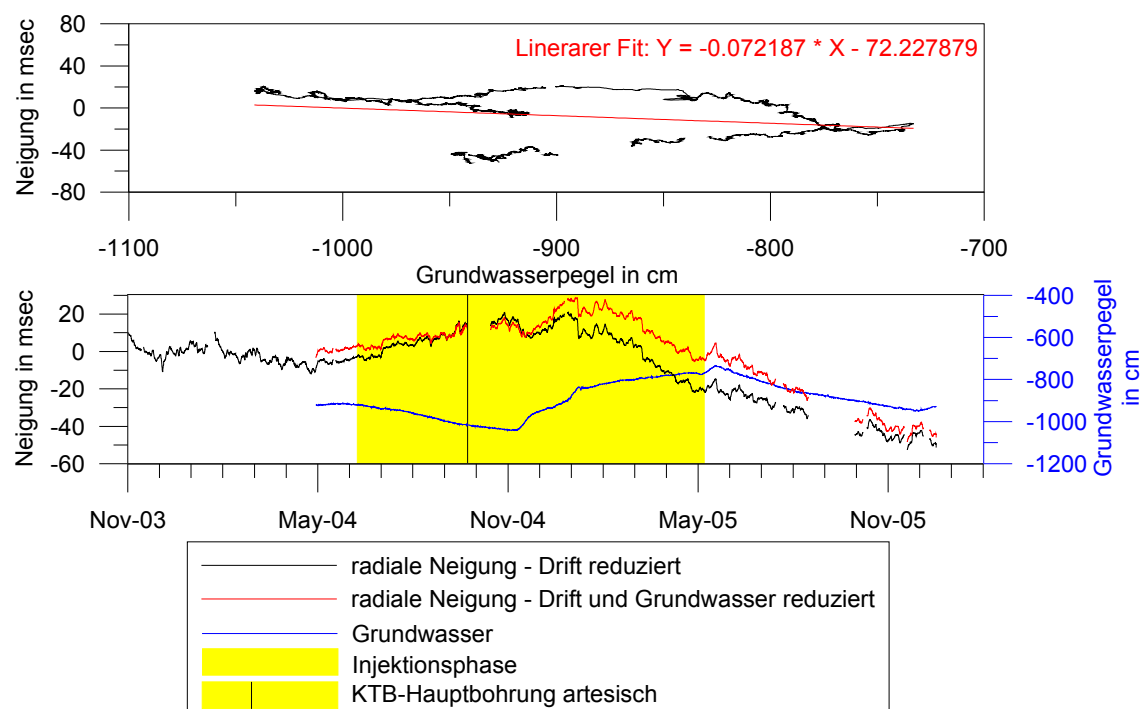


Abb. 8.10: Drift und Grundwasserreduktion der radialen Komponente von Berg.

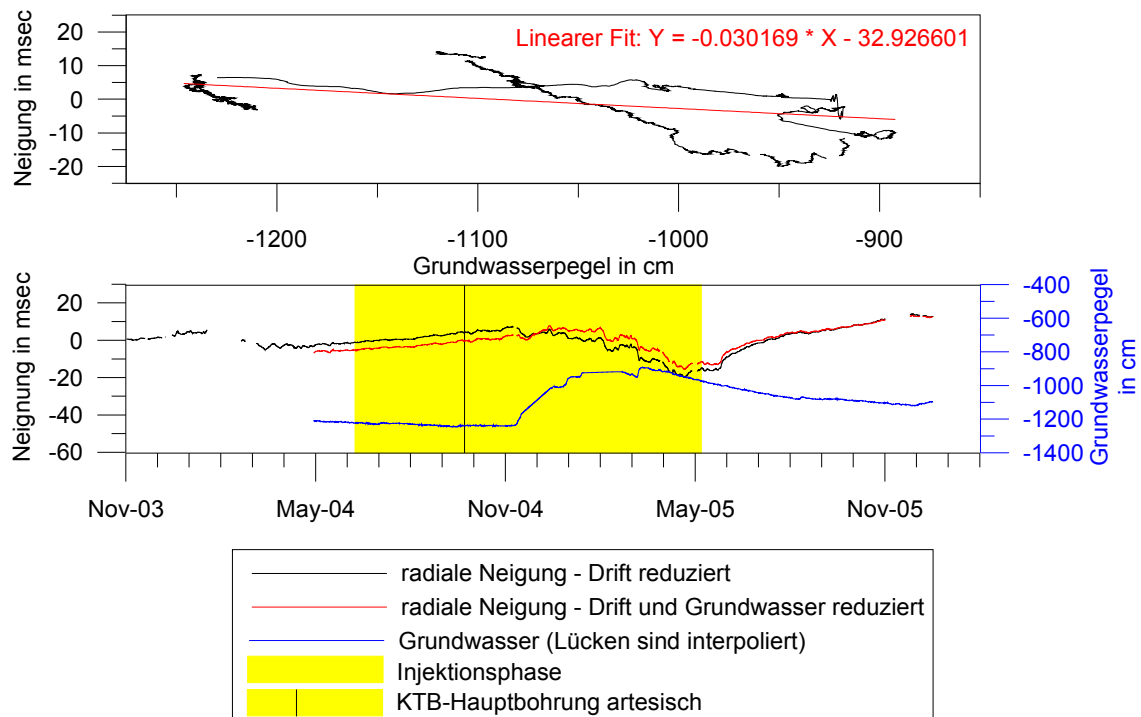


Abb. 8.11: Drift und Grundwasserreduktion der radialen Komponente von Eiglasdorf.

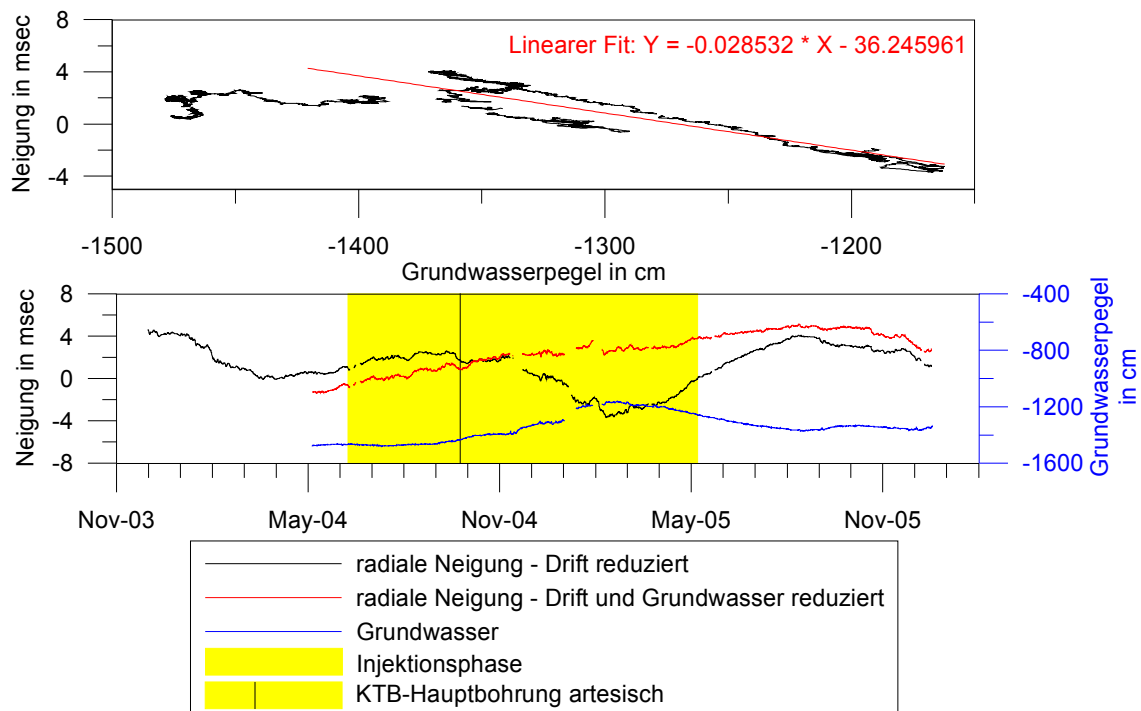


Abb. 8.12: Drift und Grundwasserreduktion der radialen Komponente von Mittelberg.

8.4 Luftdruck

Der barometrische Luftdruck spielt bei gravimetrischen Zeitreihen sowie bei Neigungsmessungen eine Rolle. Darum ist es angebracht, den Luftdruck ins Verhältnis zu den Zeitreihen der Neigungen zu setzen (Abb. 8.13). Weil die Station Mittelberg die am wenigsten gestörten, also besten Zeitreihen geliefert hat und die wenigsten Grundwassereffekte zeigt, ist der Vergleich zunächst hier am sinnvollsten. Durch die Geologie kann es durchaus möglich sein, dass der Luftdruckeffekt an den anderen Stationen in ähnlicher Weise in Erscheinung tritt.

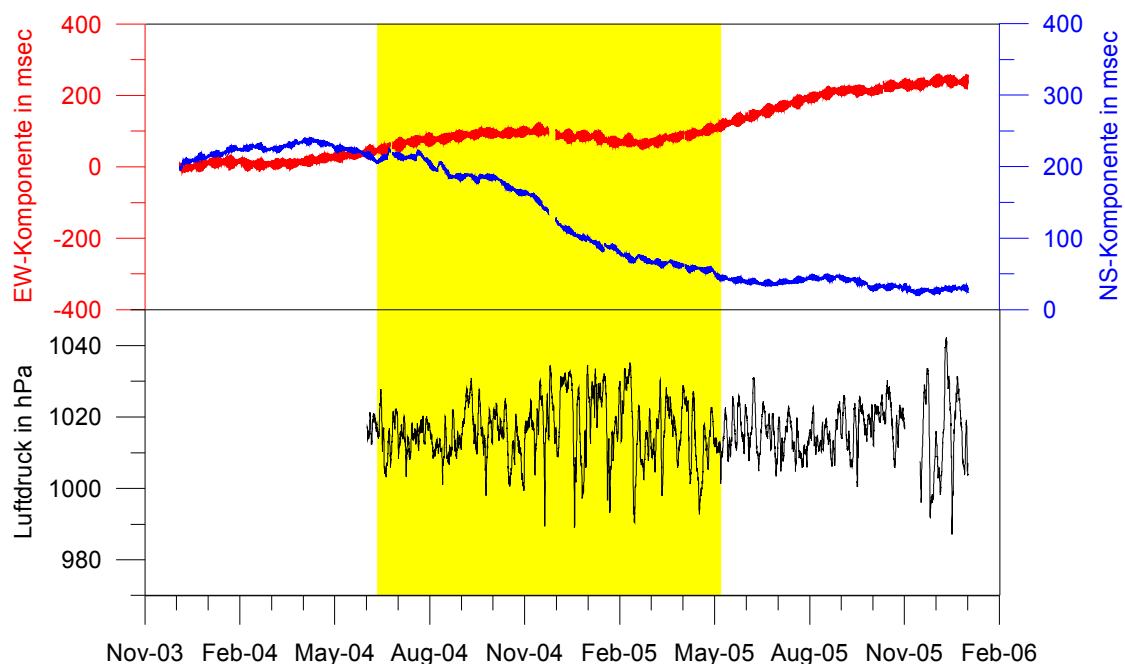


Abb. 8.13: Vergleich der Neigungen von Mittelberg mit dem Luftdruck.

In der Analyse der Gezeitenparameter mit dem Programm *Analyze* (Wenzel, 1996; siehe auch Kapitel 6.2.5) lässt sich der Luftdruck einbeziehen. Die ermittelten Koeffizienten geben die Korrelation des Einflusses der Gezeiten im Luftdruck mit den Neigungen der Gezeiten wieder. In der Ost-West-Richtung beträgt diese 0.68 msec/hPa mit einer Standardabweichung von 0.13. In Nord-Süd-Richtung ist sie um eine Größenordnung geringer, da die Hoch- bzw. Tiefdruckgebiete vorwiegend in Ost-West-Richtung ziehen und liegt bei -0.09 msec/hPa (Standardabweichung 0.06).

Vor der Betrachtung von einzelnen Luftdruckereignissen ist es notwendig die Gezeiten aus den Stundendaten (Abb. 8.14, oben) zu eliminieren. Zu diesem Zweck wur-

den theoretische Gezeiten mit den ermittelten Gezeitenparametern gerechnet und abgezogen (Abb. 8.14, 2. v. oben).

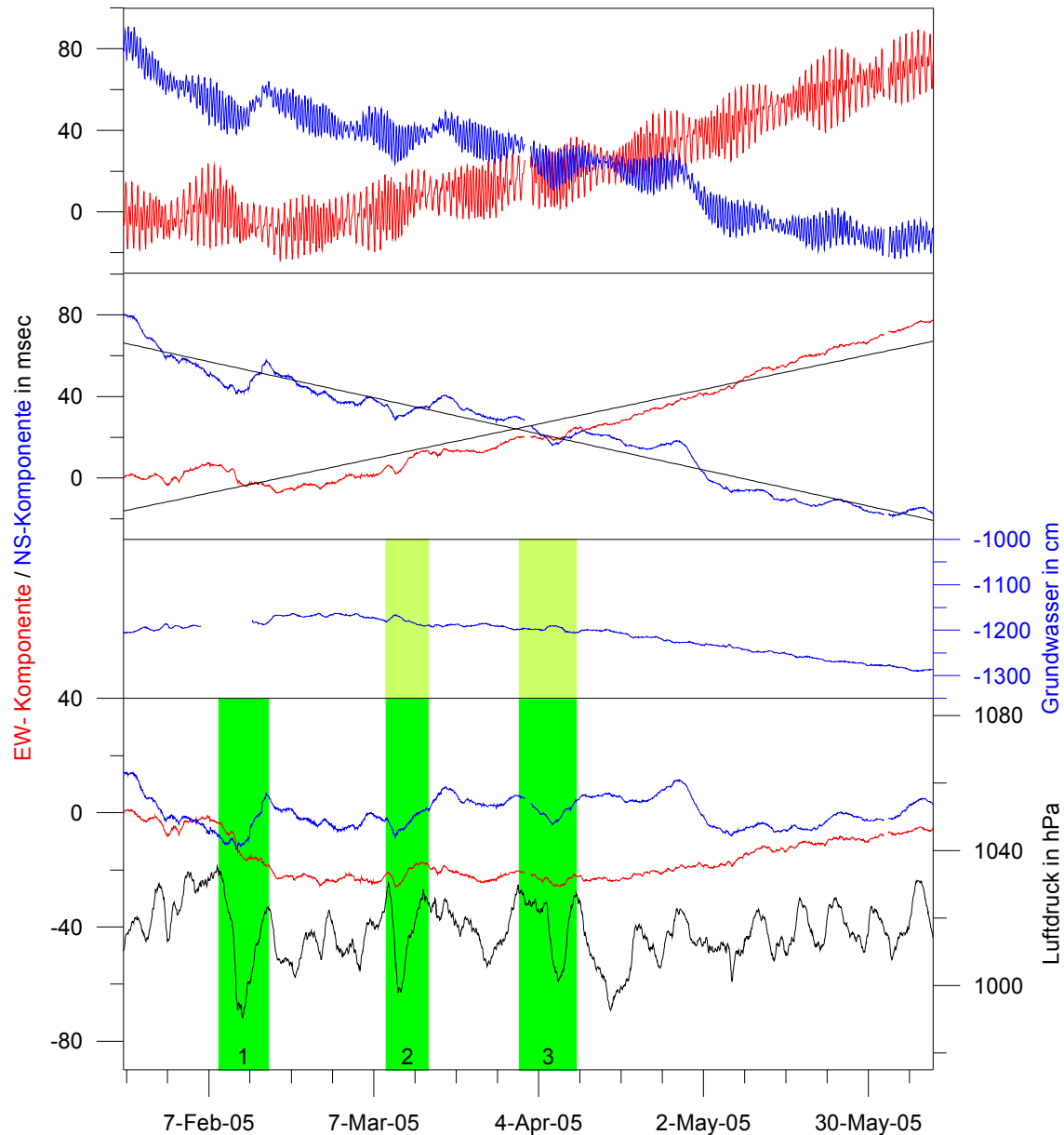


Abb. 8.14: Vergleich einzelner Luftdruckereignisse mit Neigungseffekten der Station Mittelberg.

Bedingt durch die Driften sind die hervorgerufenen Neigungseffekte nur schlecht erkennbar, darum wurde der Trend mit einer linearen Funktion approximiert und entfernt (Abb. 8.14, mitte). Der Vergleich (Abb. 8.14, unten) mit den Luftdruck zeigt nun die Abhängigkeiten. Drei einzelne Luftdruckereignisse sind in Abbildung 8.14 (unten) markiert. Das erste Ereignis (Abb. 8.14, unten (1)) vom 09.02.2005 bis zum 17.02.2005 zeigt nicht den Effekt in den Neigungen wie das zweite und dritte. Das

zweite markierte Ereignis (10.03.2005 – 17.03.2005) zeigt bei genauer Betrachtung einen zeitlichen Versatz zwischen den Maxima der Neigung und Luftdruck. Die maximale Neigungsamplitude befindet sich zu dem Zeitpunkt des größten Gradienten in der Luftdruckkurve. Im Grundwasser ist ebenfalls eine Reaktion (Abb. 8.14, 2. v. unten) zusehen. Mit der Temperatur, welche in dem entsprechenden Zeitraum sowie eine Woche davor zwischen 0° und -12°C variierte, kann der Effekt nicht durch Regen erzeugt worden sein. Möglicherweise entsteht er aber durch Tauen des Untergrundes bei einer höheren Temperatur, da die meteorologische Station die Temperatur im Schatten registriert.

Tabelle 8.2 zeigt die aus den Kurven (Abb. 8.14, unten) ermittelten Werte der maximalen Differenzen Δ der Neigungen und des Luftdruckes sowie deren Quotienten. Der zweite markierte Luftdruckeffekt zeigt eine größere Differenz in der NS-Komponente und der dritte in der EW-Komponente, der Quotient verhält sich dementsprechend. Der Unterschied kann aus unterschiedlichen Zugrichtungen des Tiefdruckgebietes erklärt werden.

Tab. 8.2: Neigungseffekte in Abhängigkeit vom Luftdruck.

Zeitraum		$\Delta\alpha_{EW}$ EW-Komponente in msec	$\Delta\alpha_{NS}$ NS-Komponente in msec	Δp Luftdruck in hPa	$\frac{\Delta\alpha_{EW}}{\Delta p}$ in msec/hPa	$\frac{\Delta\alpha_{NS}}{\Delta p}$ in msec/hPa
Beginn	Ende					
10.03.05	17.03.05	6.26	7.70	31.45	0.199	0.245
01.04.05	11.04.05	7.58	5.39	29.30	0.289	0.205

Abschätzungen von Herbst (1976) haben gezeigt, dass bei einer Luftdruckänderung von 50 hPa durch ein Tiefdruckgebiet eine Neigungsänderung bei ca. 5 msec liegt. Die Größe des Einflusses hängt von der Installationstiefe des Messgerätes aber auch von der Topographie und der geologischen Situation ab.

In den verwendeten Zeitreihen von Kapitel 8.3 und 8.5 sind die eben beschriebenen Effekte nicht mehr vorhanden, da sie bei der Filterung eliminiert wurden. Das Jahresmittel des Luftdrucks dieser Breiten hat eine Variation von 1 hPa. Damit hat der Luftdruck aus den ermittelten Werten für durchziehende Tiefdruckgebiete über das Jahr hin einen Einfluss auf die Neigung in der Größenordnung von ca. 0.3 msec.

8.5 Hodogramme

Hodogramme sind Darstellungen der Bewegung der Pendelspitze über Grund. In Abbildung 8.15 wurden die Hodogramme mit den Stundendaten (Abb. 8.2) erstellt. Deutlich zu sehen sind die 2-wöchigen Schwebungen der Gezeiten. Überlagert sind diese von Driften (Kapitel 8.3) zunächst unbekannter Ursache. Allerdings orientieren sich die Richtungen an geologischen Strukturen und möglicherweise auch am regionalen Stressfeld (mehr dazu in Kapitel 9). In den Hodogrammen sind der Anfang, das Ende, die Injektionsphase und der Zeitpunkt markiert, an dem der Hauptbohrung der KTB artesisch wurde. Etwa zu den Zeitpunkten, welche im Zusammenhang mit der Injektion stehen, ändert sich die Richtung der Driften. (Abb. 8.15, Stationen Berg, Eiglasdorf und Mittelberg). Driftrichtungen, welche mit geologischen Strukturen zusammenfallen, sind markiert (siehe dazu Kapitel 3).

Die Zeitreihen der Station Püllersreuth sind, wie schon beschrieben, sehr stark von Grundwassereffekten geprägt, daher sind keine Vorzugsrichtungen zu erkennen. Die Station Stockau machte immer wieder technische Probleme. Deshalb werden diese beiden im Folgenden nicht weiter betrachtet (ab Abbildung 8.16).

Nach Abzug der großen Driften mittels mathematischer Funktionen (Kapitel 8.3) werden weitere Informationen in den Hodogrammen (Abb. 8.16) deutlich. Bezogen auf die KTB haben sich die Driftrichtungen der Stationen Berg, Eiglasdorf und Mittelberg radialsymmetrisch ausgerichtet.

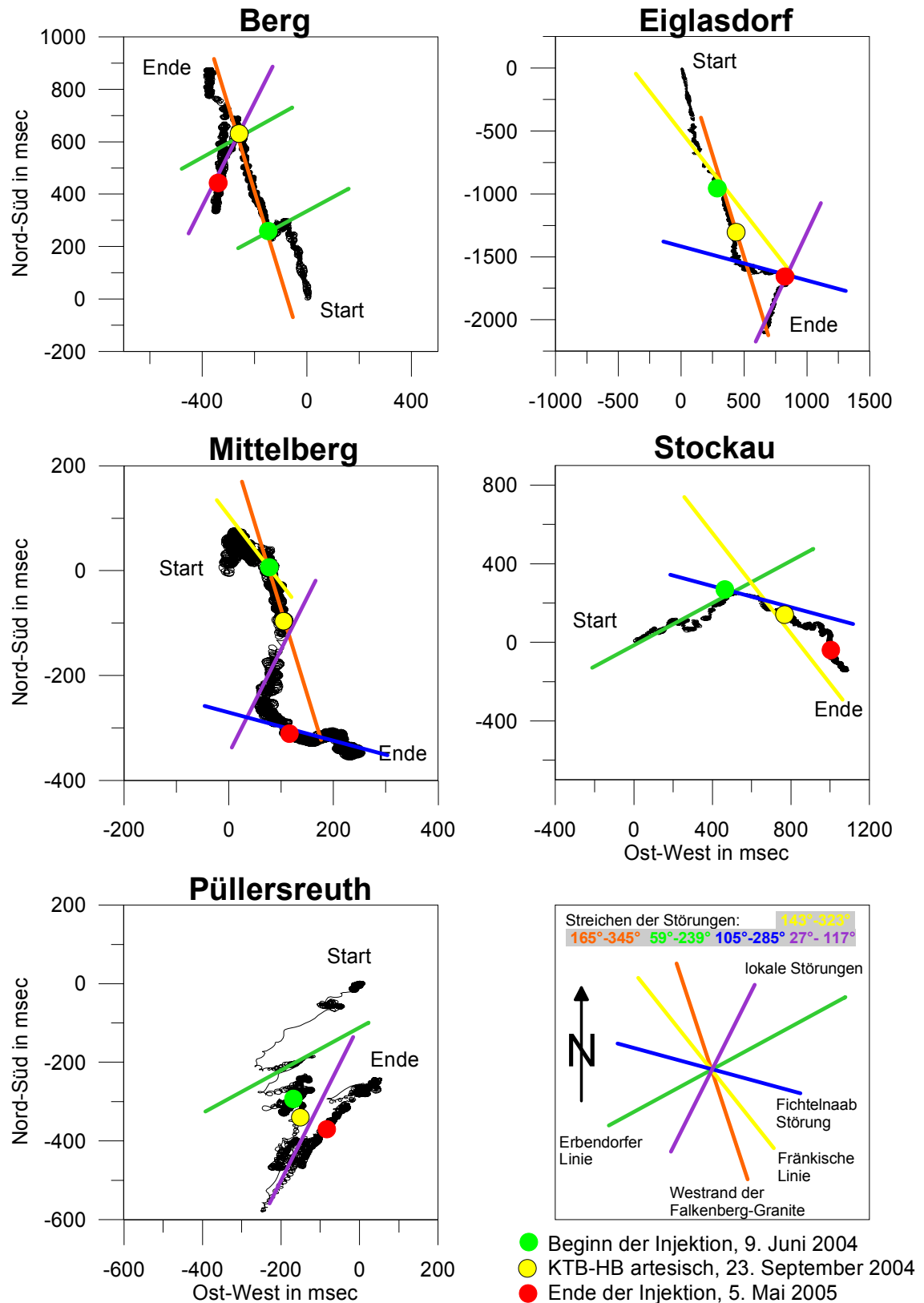


Abb. 8.15: Hodogramme der Stundendaten (Abb. 8.2).

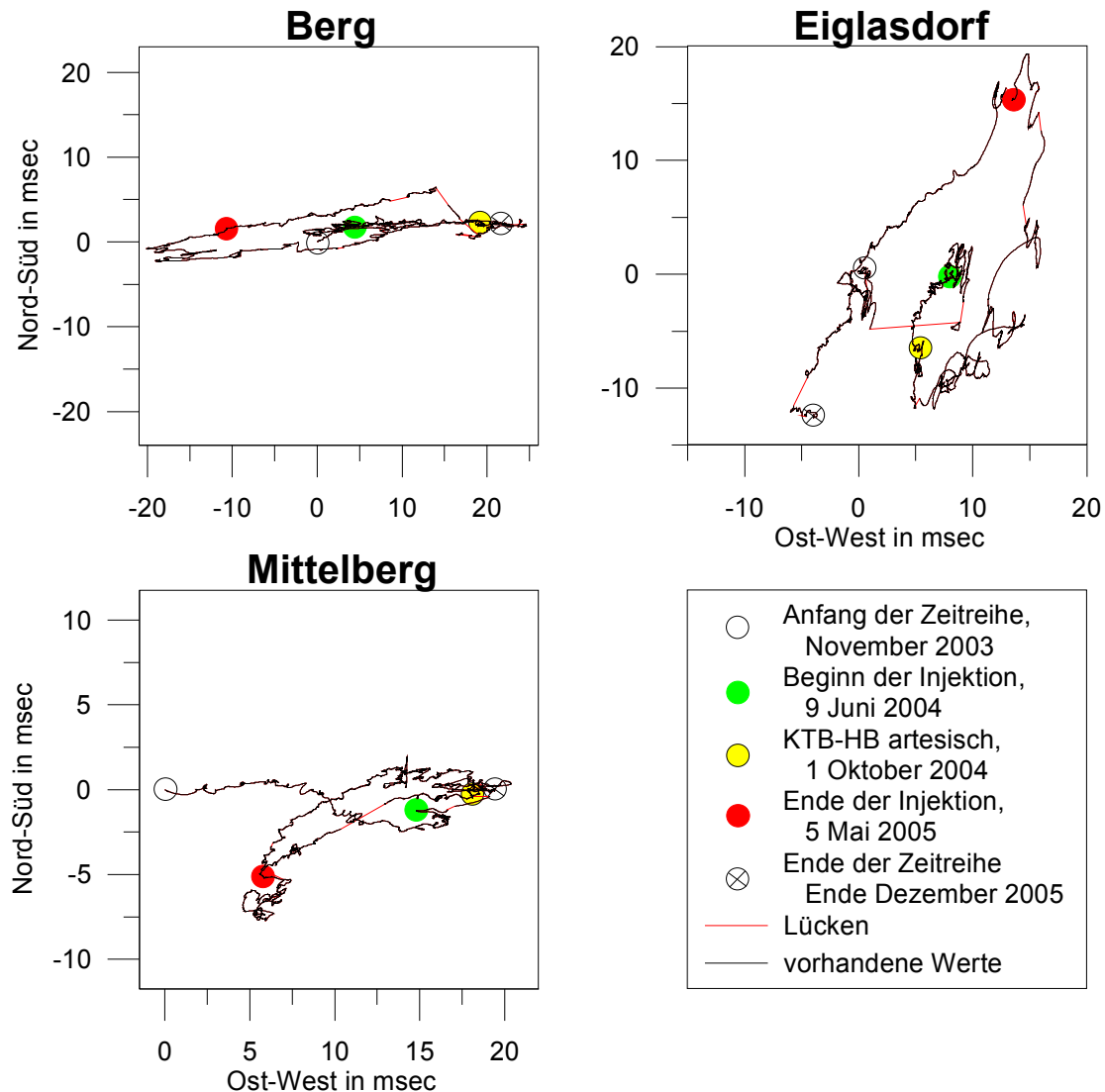


Abb. 8.16: Hodogramme der Drift-reduzierten und der zusätzlich langperiodisch gefilterten Stunden-daten (aus Abb. 8.9).

Die zeitlichen Eckpunkte der Injektion fallen wiederum mit Richtungsänderungen der Graphen in den Hodogrammen zusammen (Abb. 8.16).

In Abbildung 8.18 sind die drei Stationen Berg, Eiglasdorf und Mittelberg nochmals aufgegriffen worden. Hier wurden die Drift-reduzierten Hodogramme, den zusätzlich Grundwasser-reduzierten (Kapitel 8.3) gegenüber gestellt. Mit Abbildung 8.17 sind die erwähnten Driftrichtungen ersichtlich. Die Ursache der leichten Drehung ist noch zu klären. Die Pendelspitze der drei Neigungsmesser wandern radial nach außen weg, nachdem die Hauptbohrung artesisch wurde. Dies bedeutet, dass die Oberfläche in der Art deformiert wird, wie es im Kapitel 1.1 erwartet wurde.

Auffällig ist (Abb. 8.18), dass selbst nach Abzug des Grundwassereffektes die Driftrichtungen erhalten bleiben. Dabei ist zu erwähnen, dass das Grundwasser- und das Neigungssignal nur bei Mittelberg sehr gut korreliert. Bedingt durch die weitaus kürzeren Zeitreihen der Grundwasserstände (Beginn 1 Monat vor Injektion) sind in den nur Drift-reduzierten Hodogrammen die fehlenden Werte wie Lücken markiert.

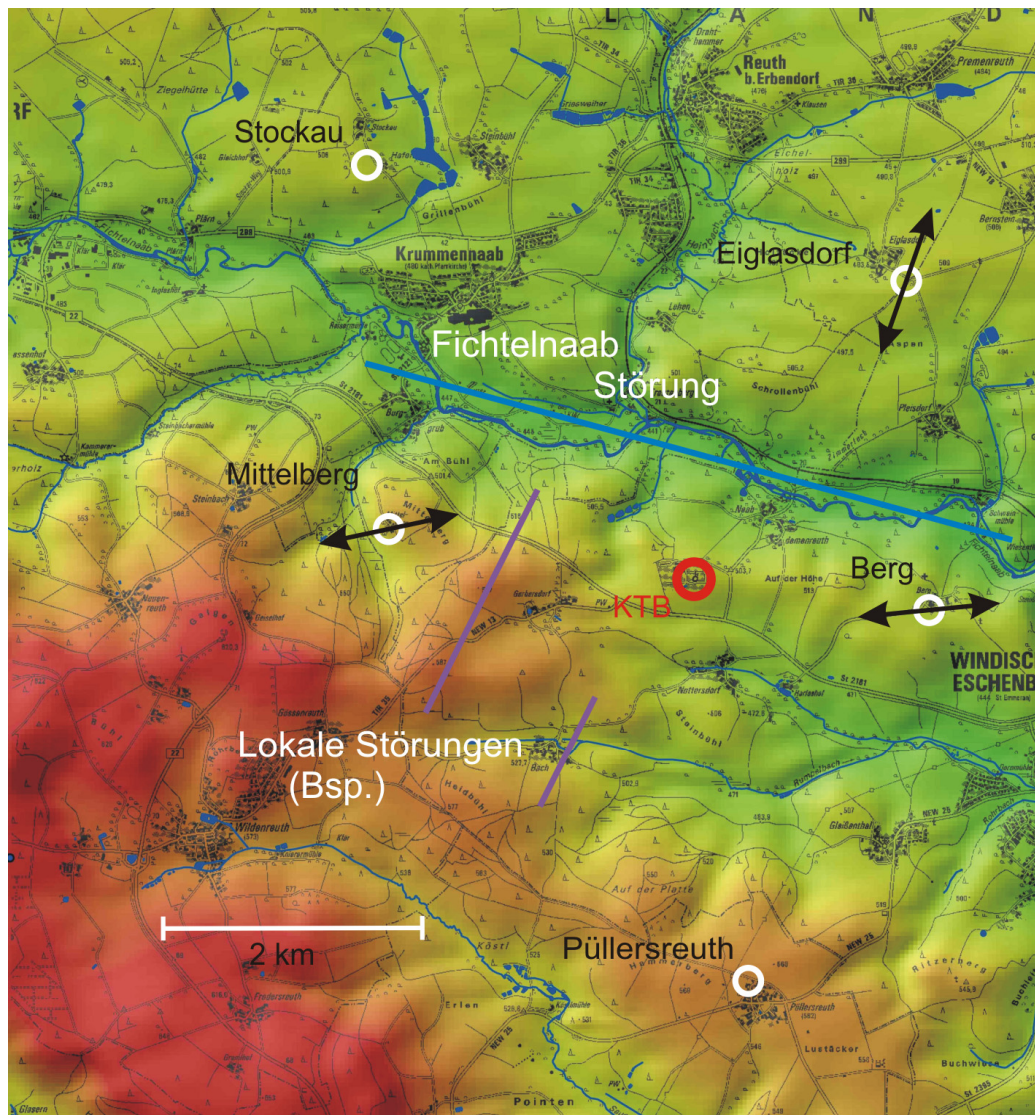


Abb. 8.17: Lage der Stationen den mit lokalen Störungen und den Driftrichtungen der Hodogramme aus Abbildung 8.16.

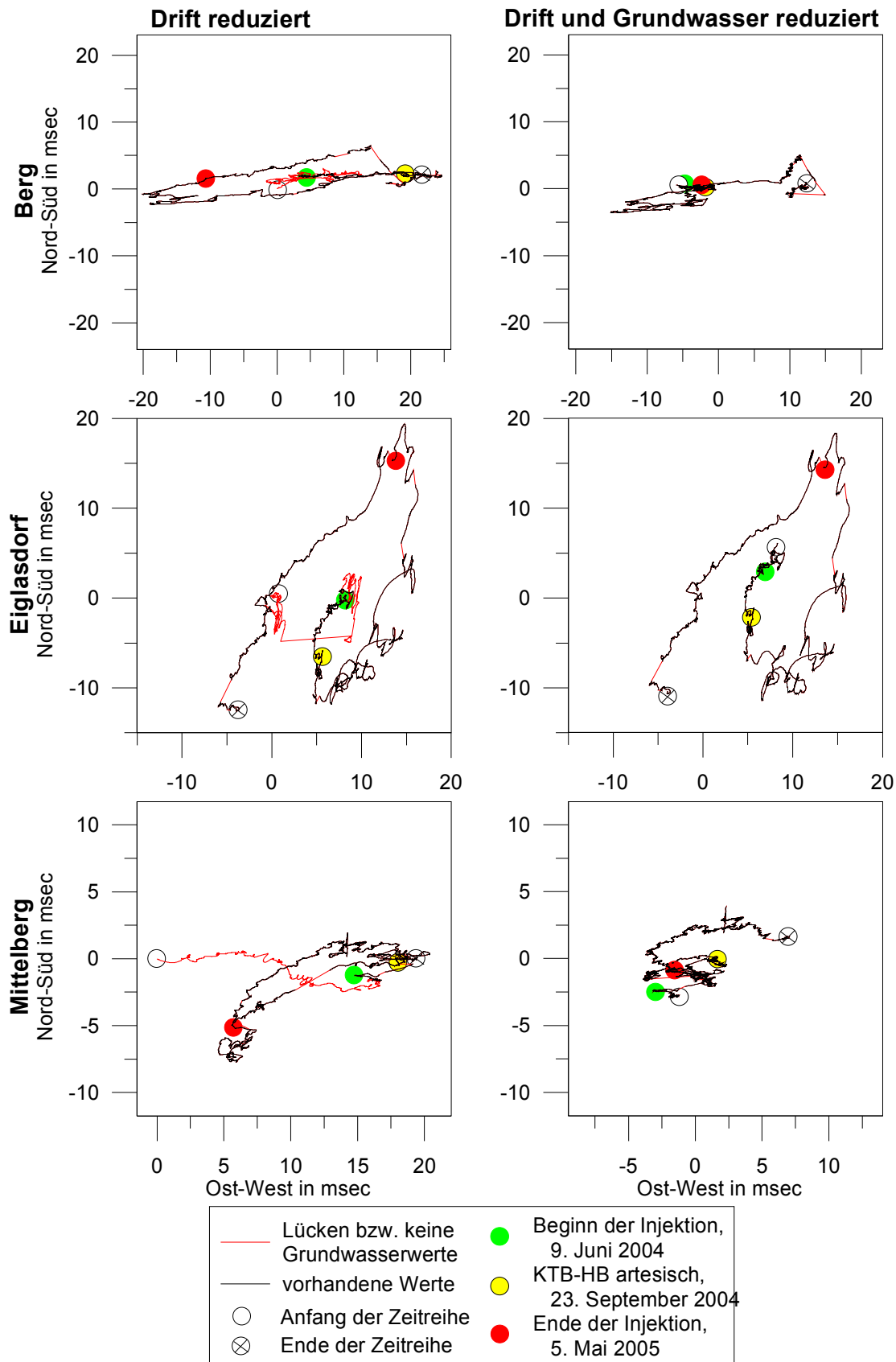


Abb. 8.18: Hodogramme der Drift- und Grundwasser-reduzierten langperiodisch gefilterten Stunden-

8.6 Das Sumatra-Andamanen-Beben vom 26.12.2004

Am 26.12.2004 um 00:58:00 Herdzeit (UTC) fand vor Sumatra ein Beben statt, dessen Magnitude mit $M_w = 9.3$ ermittelt worden ist. Die ausgelösten Wellen erreichten um 01:11 UTC die Neigungsmesser. Die maximale Doppelamplitude der daraus verursachten Neigungen betrug etwa 2000 msec (Abb. 8.19, oben). Zusätzlich wurden starke Erdeigenschwingungen ausgelöst. Von Ihnen gibt es verschiedene Arten: spheroidale und toroidale. In Schwereregistrierungen wurden sie noch Monate nach dem Beben beobachtet. Die durch die Erdeigenschwingungen ausgelösten horizontalen Beschleunigungen verschwanden schon nach einigen Tagen im Rauschen der Zeitreihen. In Abbildung 8.19 (Mitte) sind diese deutlich in der EW-Komponente der Station Mittelberg als Modulation auf den Gezeiten mit einer Periode von größer als 3.5 Minuten zu sehen.

Für die einzelnen Komponenten der Stationen wurden Spektren gerechnet (Jentzsch et al., 2005). Optimale Ergebnisse lieferten Stücken von zwei Tagen Länge vom 26.12.04 6:00 Uhr UTC bis zum 28.12.04 6:00 Uhr UTC. Hierzu ist anzumerken, dass es sich bei den Erdeigenschwingungen um freie gedämpfte Schwingungen handelt, deren einzelne Moden unterschiedlich schnell im Rauschen verschwinden. Mit dem Neigungsmesser-Array bot sich die bislang einzigartige Möglichkeit, Neigungsbeobachtungen zu stapeln. Geschehen ist dies mit den Spektren der jeweiligen Komponenten (Abb. 8.19, unten), dadurch hat sich das Signal-Rausch-Verhältnis stark verbessert. Den spektralen Amplituden sind die Ursachen zugeordnet; beobachtet werden toroidale und spheroidale Moden, die Neigungsänderungen erzeugen.

Erdeigenschwingungen werden seit dem großen Chile-Erdbeben von 1960 intensiv diskutiert. Eine recht umfassende Darstellung der elastischen Erdeigenschwingungen geben Zürn und Widmer-Schniedrig (2002), in der auch die Grundmode ${}_0T_2$ mit ihrer Frequenz von 377.3 μHz (44.2 Minuten) enthalten ist, die bislang nur sehr selten separiert werden konnte (Widmer et al., 1992). Diese Mode ist deutlich im Spektrum zu sehen. Flach & Große-Brauckmann (1974) diskutierten zum ersten Mal Erdeigenschwingungen, die mit einem Askania-Bohrloch-Neigungsmesser beobachtet worden waren.

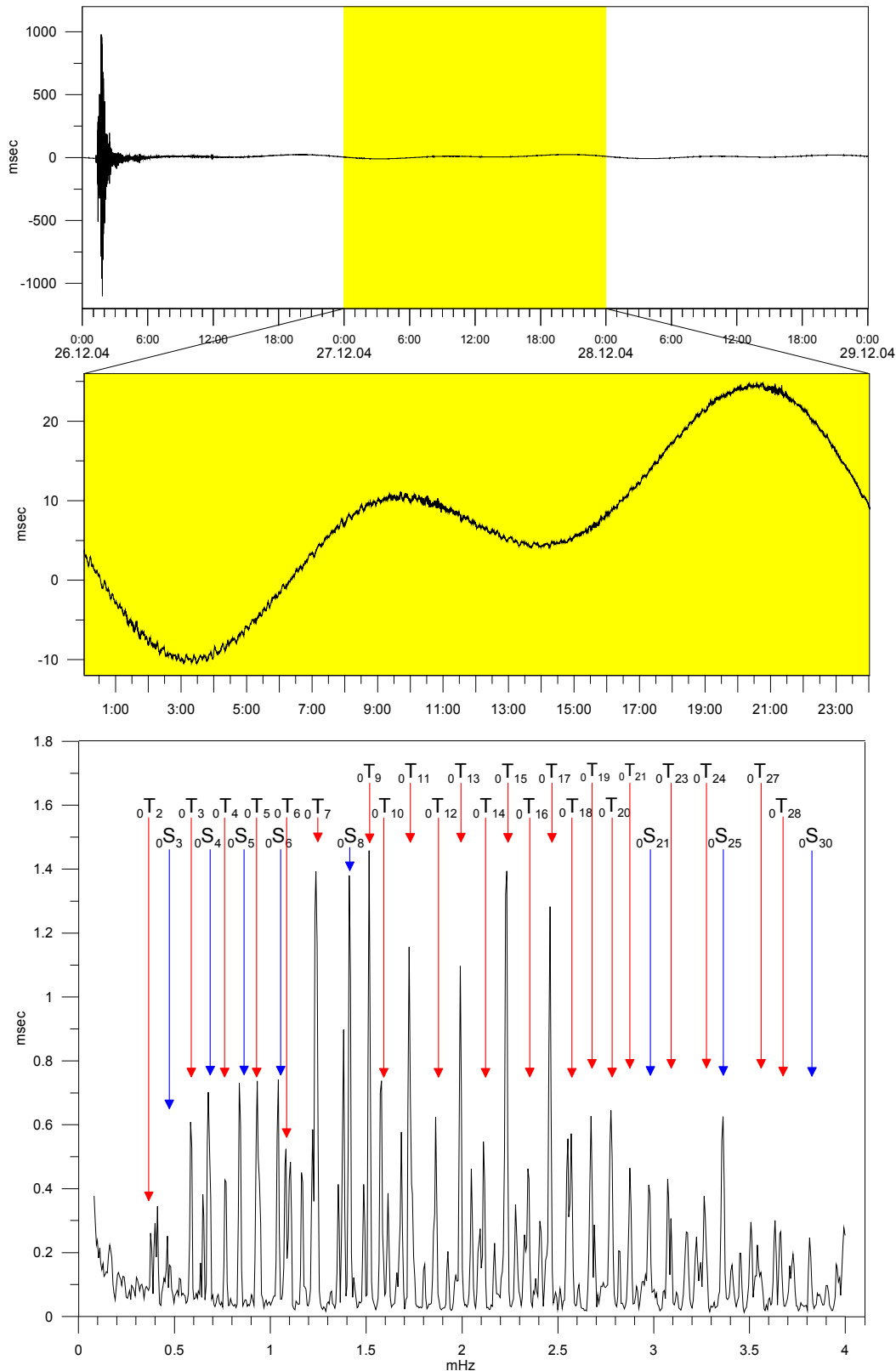


Abb. 8.19: Oberflächenwellen des Sumatra-Andamanen-Bebens vom 26.12.2004 – EW-Komponente Mittelberg (oben), Erdeigenschwingungen auf Gezeitenwellen – EW-Komponente (Mitte), Spektrum der gestapelten EW-Komponente der fünf Stationen für 2 Tage – 26.12.04 6:00 UTC bis 28.12.04 6:00 UTC (unten).

9 Interpretation und Diskussion

Die sich in Kapitel 8 andeutenden Abhängigkeiten bzw. Wechselwirkungen der in dem Injektionsprojekt vorhanden Größen wird in Abbildung 9.1 schematisch dargestellt. Der Ansatz, welcher sich aus der Injektion und den grundlegenden Messungen der Neigung und des Porendruckes zusammensetzt, ist im Kapitel 1 beschrieben, der physikalische Hintergrund im Kapitel 4 und der Verlauf der Injektion in Kapitel 7. Es sollen nun die Einflussfaktoren sowie die Datenbearbeitung betrachtet werden.

Die als Pfeile in Abbildung 9.1 gekennzeichneten Wege zeigen die Abhängigkeiten auf. Die Abstufung durch die unterschiedlichen schwarzen Linien soll ein Hinweis darauf sein, dass die Abhängigkeiten nur schwer bzw. gar nicht zu fassen sind. Der Einfluss des Messinstrumentes bzw. Registriersystems wird zwar versucht, technisch so gut wie möglich auszuschalten, ist aber nicht vollständig eliminierbar.

Nicht zu vergessen ist der Einfluss der Temperatur, die Teil eines jeden Faktors ist. Als saisonale Welle wird sie in gewisser Weise im Luftdruck, Niederschlag, Grundwasser, den Messinstrumenten und vielleicht sogar in der Geologie zu finden sein. Dieser Gesichtspunkt führt zum ersten Problem: Vor der Injektion fand ein einjähriger Pumpversuch statt (Jahr et al., 2005; Sauter, 2005¹). Da zu der Zeit die Neigungsmesser installiert wurden, bedeutet dies, dass die Daten, welche 7 Monate vor der Injektion gesammelt wurden, nicht unbedingt den ungestörten Ausgangszustand widerspiegeln. Günstig für das Projekt wäre eine Registrierung der Deformationen von mindestens einem Jahr vor dem Pumpen gewesen, um mögliche Effekte mitzuerfassen, und von derselben Zeitdauer nach Abschluss der Injektion. Ein zusätzliches Problem bildet die sehr späte Installation der Grundwassersensoren und der Wetterstation. Dies erschwert das Separieren der durch die Injektion induzierten Drift.

¹ Pers. Mitteilung

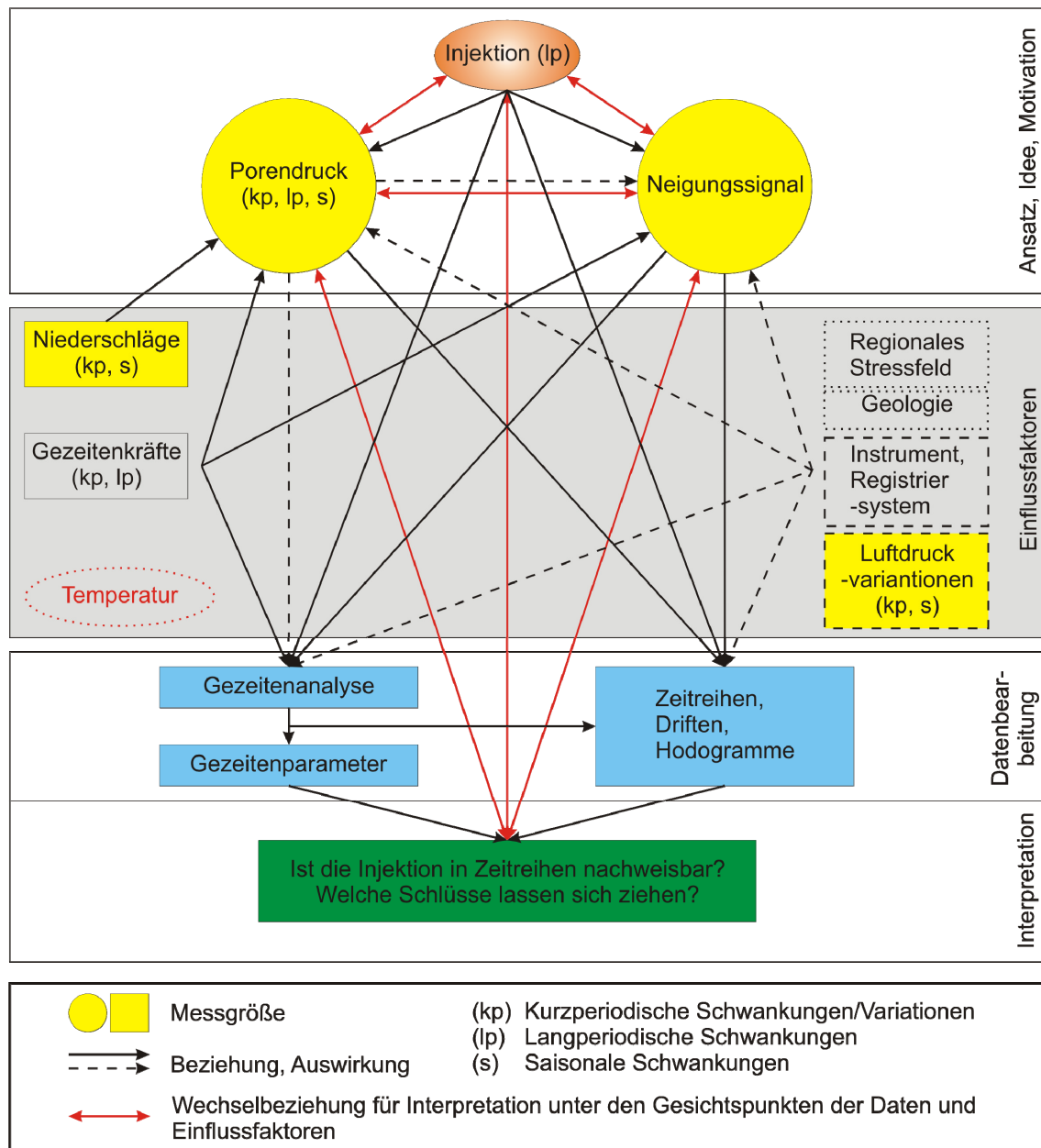


Abb. 9.1: Zusammenhänge der Messgrößen und Einflussfaktoren.

In erster Linie ist zu klären, ob die Injektion in den Zeitreihen nachweisbar ist. Um Aussagen treffen zu können, wurde aufgrund der Größenordnung der erwarteten Neigungen entsprechend der beschriebenen Vorgehensweise versucht, alle bekannten Elemente und Einflussfaktoren zu eliminieren bzw. zu reduzieren oder abzuschätzen und schlussendlich zu separieren. Bevor dies im Folgenden diskutiert werden soll, werden nochmals kurz die durch die Injektion ausgelösten Effekte dargestellt.

Injektion

Die zwei grundlegenden Effekte, welche durch eine Injektion ausgelöst werden, sind schematisch in Abbildung 9.2 dargestellt. Zum einen wird der Grundwasserspiegel lokal angehoben und zum anderen wird der Untergrund deformiert.

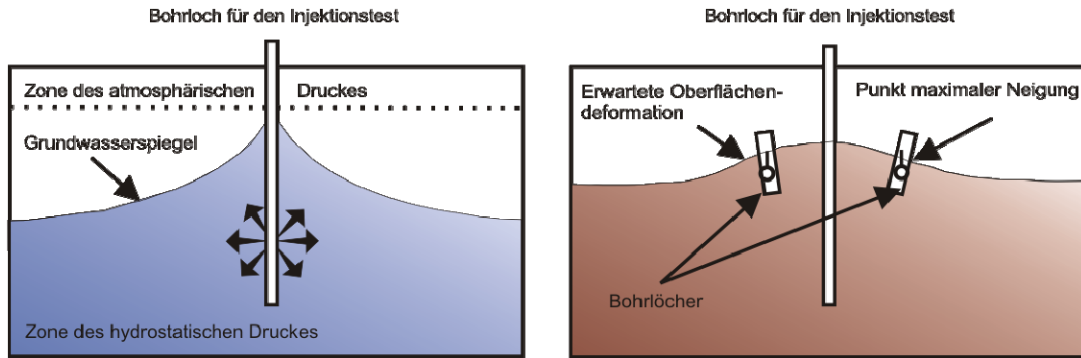


Abb. 9.2: Schematisierte Auswirkungen von Injektionen.

In der Tiefe des Injektionspunktes von vier Kilometern kann davon ausgegangen werden, dass die Porenräume vollständig mit Fluid gefüllt sind und somit nur der Druck erhöht wird. Nähert man sich dem Grundwasserspiegel (auf wenige Meter), so besteht die Möglichkeit, dass noch nicht gefüllte Poren gefüllt werden und dadurch der Spiegel angehoben wird. Darum ist die erste zu klärende Frage, ob sich in den Grundwasseraufzeichnungen ein Signal der Injektion extrahieren lässt.

Durch das eingebrachte Fluidvolumen und die Druckerhöhung wird die Oberfläche bei homogenem, isotropem Untergrund radialsymmetrisch verformt. Aus Vorabschätzungen und Modellierungen wurde für die Installation eine ideale Entfernung von 2.4 km mit einem Neigungseffekt von etwa 5 msec bestimmt. In die Vorbetrachtungen, die aus einer einfachen 3D Finite-Element Modellierung besteht, sind Parameter eingegangen, die mit Hilfe von vorherigen Versuchen an der KTB bestimmt wurden, wie zum Beispiel der Durchlässigkeitsbeiwert (K_f -Wert). Dieser Wert war, wie sich durch die Injektion herausstellte, zu klein. Ausgewirkt hat sich dies in den während der Injektion erreichten Drücken, welche geringer waren als angenommen.

Mit der Gleichung (9.1) (auch Gl. 4.41)

$$\Delta_s = -M^* \frac{3rz}{R^2} \frac{(3\mathfrak{I} + \epsilon^-)}{4} \quad (9.1)$$

lässt sich abschätzen, welche Auswirkung der K_f -Wert (Gl. 9.2) auf die zu erwartenden Neigungen hat, wobei in der Poroelastizität die hydraulische Diffusivität D (Gl. 9.3; berücksichtigt Matrixdeformation) verwendet wird.

$$K_f = \kappa \rho_f g \quad (9.2)$$

$$D = \frac{2}{9} \frac{(1-\nu)(1+\nu_u^2)}{(1-\nu_u)(\nu_u-\nu)} \kappa \mu B^2 \quad (9.3)$$

Mit den Substitutionstermen $M^* = \frac{Q_f \bar{B}}{4\pi R D}$ und $\mathfrak{Z} = \frac{\epsilon / \mathfrak{R} - \dot{\epsilon}}{2\mathfrak{R}}$ sowie dem dimensionslosen Abstand $\mathfrak{R} = \frac{R}{2\sqrt{Dt}}$ von der Punktquelle, lässt sich mit der Bedingung, dass die Errorfunktion ϵ den maximalen Wert 1 annehmen kann, die Gleichung 9.1 auf $\Delta_s \sim \frac{1}{\sqrt{Dt}}$ abschätzen. Das bedeutet, dass mit einem größeren als dem erwarteten D (K_f) die Neigungskomponente der Deformationen bei gleicher Injektionsdauer kleiner wird.

In Kapitel 7 wurde die Injektion beschrieben. Der Kopfdruck der Vorbohrung fiel durch die Abkühlung des Gebirges zu Beginn ab und stieg dann langsam wieder an. Ob dieser Effekt einen Einfluss auf die Neigungen und Pegelstände hat, lässt sich nur durch eine Modellierung zeigen. Zu dem Zeitpunkt, als die Hauptbohrung artesisch wurde, fällt der Druckzuwachs bei andauernder Injektion durch das freie Ausfließen auf das Gebirge weg. Der steigende Ausfluss in Verbindung mit dem langsam jedoch stetig steigenden Druck am Kopf der Vorbohrung ist ein Zeichen dafür, dass sich trotzdem am Injektionspunkt der Druck erhöht.

Dies kann wiederum nur durch eine Modellierung beantwortet werden, wobei die bestimmenden Parameter Durchlässigkeit und die Struktur des Untergrundes sind.

Grundwasser und Niederschlag

Zur Klärung der in den obigen Abschnitten aufgeworfenen ersten Frage muss man sich die Daten der Grundwasserpegel, der Temperatur und des Niederschlags ansehen (Abb. 9.3). Harrison und Herbst (1977) haben gezeigt, dass durch Temperaturschwankungen thermoelastische Spannungen auftreten, welche auch stark von der Topographie abhängen. Aus Modellrechnungen ist hervorgegangen, dass ab einer Tiefe von 35 m diese Neigungseffekte vernachlässigbar sind. Mit den Ausführungen der obigen Abschnitte und denen aus Kapitel 8 lässt sich die Vermutung äußern, dass die Wasserstände bei den Stationen Berg, Eiglasdorf und Mittelberg in Bezug auf die Injektion reagiert haben, die

von Stockkau und Püllersreuth aber nicht. Eindeutig lässt sich dies schon alleine aus dem Grund nicht sagen, da die beobachteten Zeitreihen zu kurz sind. Es fehlt hier mindestens ein Jahr ungestörter Aufzeichnungen, um den saisonalen Anteil abschätzen zu können. Die Vermutung stützt sich auf die phasenverschobenen Extrema der jährlichen Variation (Abb. 9.3, unten, rote Linien) und auf die Größe der Amplituden, sowie die nicht vorhandenen Niederschläge, welche auch als Schnee gefallen sein können, aber nicht gemessen wurden. Dabei sind lokale Effekte nicht zu vernachlässigen.

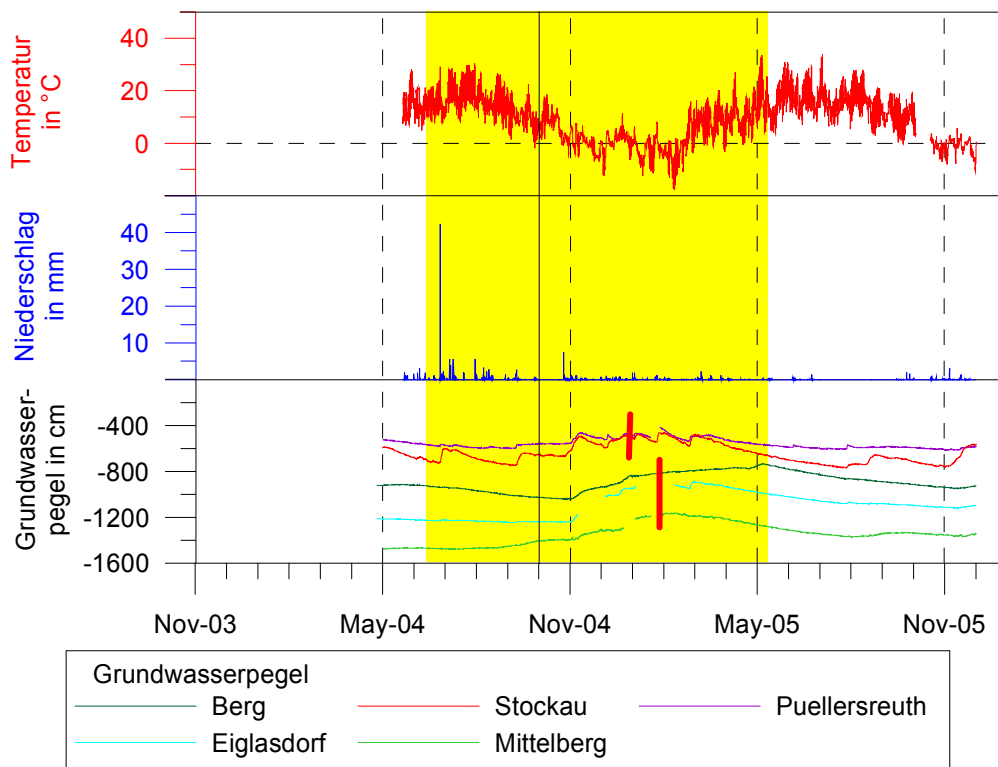


Abb. 9.3: Vergleich der Grundwasserpegelstände mit der Temperatur sowie dem Niederschlag.

Bei genauerer Betrachtung (Abb. 9.3) ist Anfang März 2005 bei höher werdenden Temperaturen ein deutliches Signal in den Pegeln der Stationen Eiglasdorf, Stockkau und Püllersreuth zu erkennen. Dies lässt sich durch die Schneeschmelze erklären.

Auffällig ist der Verlauf der Graphen der Station Berg, Eiglasdorf und Mittelberg, da sie alle gleichartig reagieren (Anstieg und Abfall der Pegel in der langen Periode). Zudem beginnt in Berg, der KTB nächst gelegenen Station, genau mit Ende der Injektion der Pegel langsam zu fallen. Warum dies in Eiglasdorf und Mittelberg nicht zu beobachten ist, lässt sich vielleicht durch lokale geologische Ursachen erklären.

Mit den Fragen, welcher Faktor im Winter Grundwasserpegelstände ansteigen lässt und wieso sie in einem so kleinem Gebiet im Hintergrund der Injektion so ähnlich reagieren, drängt sich unter allen Vorbehalten die Erkenntnis auf, dass sich die Injektion auf die Pegelstände ausgewirkt hat.

Um diese Vermutung zu verifizieren, würde es noch einen weiteren Ansatzpunkt geben, nämlich die Betrachtung des Volumenstrains in derselben Art und Weise wie die Variation der Gezeitenparameter der beobachteten Neigungen.

Luftdruck

Die Luftdruckschwankungen haben, wie unter Kapitel 8.4 ausgeführt, einen Einfluss auf die Neigungen und den Porendruck. Im Zusammenhang mit der Injektion ist jedoch die Größe des Einflusses interessant. Durch einzelne Tiefdruckgebiete (Kapitel 8.4) werden zwar Neigungen in der Größenordnung der alleine durch die Injektion abgeschätzten ausgelösten Neigungen induziert. Jedoch sind die durch die Variation des Luftdrucks um das jährliche Mittel verursachten Effekte kleiner als die durch die Injektion verursachten. Bedingt durch die Injektionsdauer von 10 Monaten und durch die Verwendung der gefilterten Zeitreihen der Neigungen kann dieser Effekt als vernachlässigbar betrachtet werden.

Analyse der Gezeitenparameter

Bei anhaltender Injektion kann von einer Vergrößerung der v_p -Geschwindigkeit ausgegangen werden (Kummero, 2005)¹. Da v_p quadratisch in die Berechnung des E-Moduls eingeht, wird das E-Modul durch die Injektion vergrößert, d.h. die oberste Erdkruste wird starrer. Dies bedeutet, dass die beobachteten Gezeitenamplituden der einzelnen Wellen kleiner und damit auch der Gezeitenparameter kleiner werden muss.

Der hier vorgestellten Argumentation steht aber die Beobachtung aus Schwarm-erdbebengebieten gegenüber: Durch Einwirkung von Fluiden in einen durch das regionale Spannungsfeld gespannten Krustenblock, der Störungszonen enthält, vergrößert sich die beobachtete Deformation. Hier wurde nun, bedingt durch das Fluid, eine größere Gezeitenamplitude beobachtet und der Untergrund wurde elastischer.

Abbildung 9.4 zeigt, dass auch für die Injektion an der KTB eine Zunahme des γ -Faktors und damit eine Verringerung des E-Moduls, also ein „elastischeres Verhalten“ beobachtet wird.

¹ Persönliche Mitteilung

In Kapitel 8.2 wurde unter anderem ausgeführt, dass die Qualität der Analysen stark von der Länge der verwendeten Daten-Abschnitte abhängt. Abbildung 8.5 wurde in der Hinsicht nachbearbeitet, dass Analysen zu kurzer Zeitreihen (unter 45 Tagen) in alle vier Mittelungen gleichermaßen nicht eingeflossen sind (siehe Abb. 9.4). Die Graphiken haben sich dahin geändert, dass die Anstiege der angepassten Geraden für die NS-Komponente von einem negativen zu einen positiven Anstieg gewechselt haben, während sich in der EW-Komponente so gut wie keine Änderung einstellte.

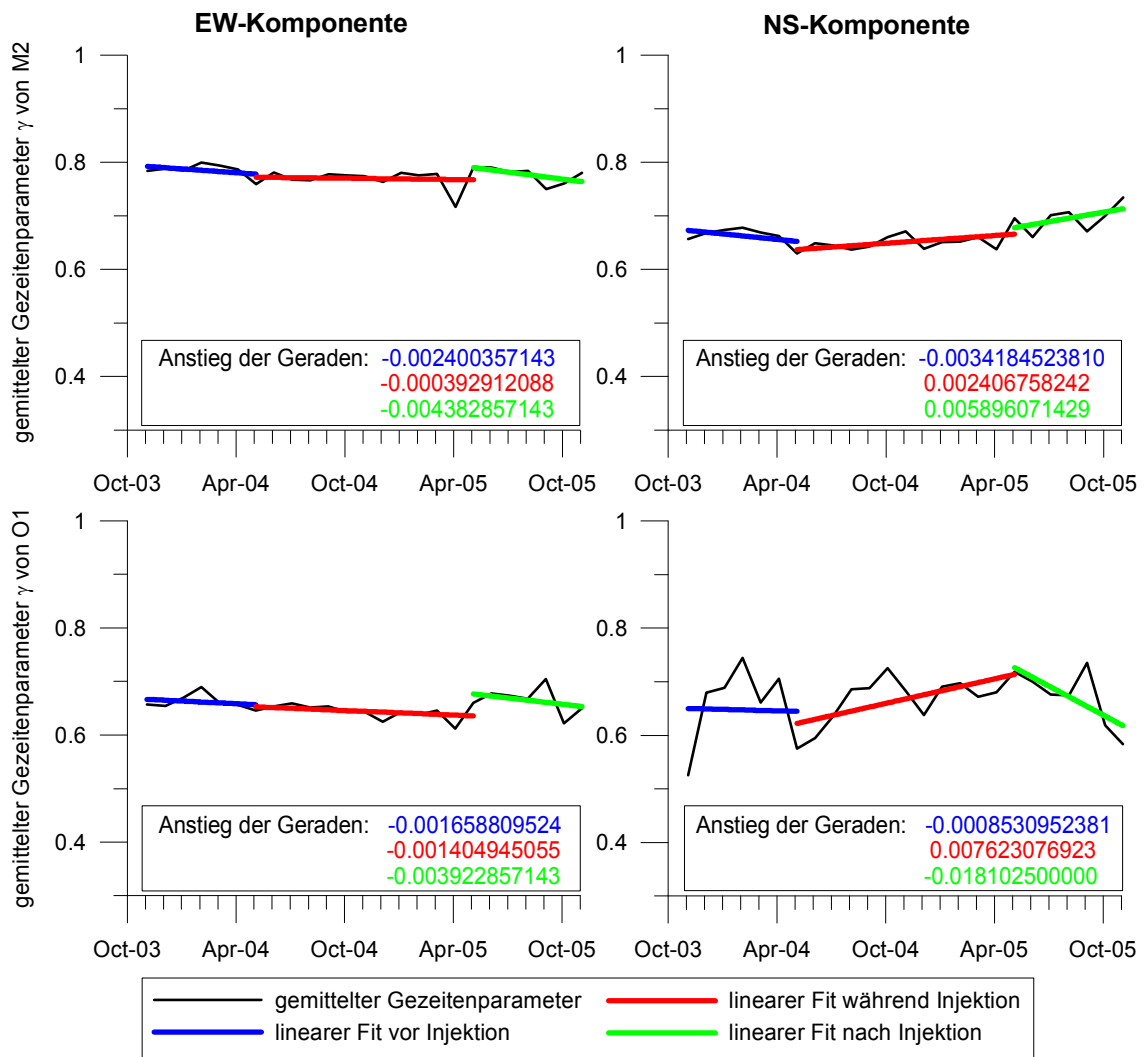


Abb. 9.4: Gemittelte Gezeitenparameter aus Analysen 2-monatiger Länge (nachbearbeitet).

Nur die Anstiege der Geraden nach der Injektionsphase der NS-Komponente der M2- und O1-Wellen differieren. Der wichtigste Punkt ist die Änderung der Richtung der Geraden mit der Injektionsphase der NS-Komponente. Unter dem Gesichtspunkt,

sich im Grenzbereich der Interpretation aufzuhalten, kann dieses Indiz auf die Widerspiegelung der Injektion in den Daten gezählt werden, da wie geschildert die γ -Faktoren größer werden.

Die angepasste Gerade der NS-Komponente der O1-Welle für den Zeitraum nach der Injektion spiegelt die Erwartungen mit dem negativen Anstieg wider. Doch warum nicht die M2-Welle? Durch den geringeren Anstieg der Geraden während der Injektion lässt sich sagen, dass die Gezeitenwelle M2 nicht so empfindlich auf die Änderung der Elastizität reagiert. Zudem sind wie schon angesprochen die Analysen trotz Überarbeitung kritisch zu betrachten. Mit Ende der Injektion ändert sich die Antwort des Untergrundes erneut.

Die stärkere Reaktion in der NS-Richtung hat sich drei Monate nach Beginn der Injektion in der Seismologie schon angedeutet. Die erwarteten Erdbeben fanden fast vollständig nördlich der Injektion statt. Die Erklärung dafür ist sicherlich in der Geologie begründet.

Drift der Station Stockau

Das Problem der Reduktion der Drift für die Station Stockau soll hier gesondert diskutiert werden, da bedingt durch Besonderheiten das Verständnis erschwert wird. Der ASKANIA-Bohrlochneigungsmesser musste im August 2004 aufgrund eines Blitzschlages ausgewechselt werden. Nachdem dies geschehen war, änderte sich die Drift-richtung (Abb. 9.5). Dies fällt aber wiederum mit dem Zeitpunkt des Injektionsbeginns zusammen (siehe auch Abb. 8.15). Nach Abzug der Driften zeigen die Zeitreihen abgesehen vom Grundwassereinfluss nur einen negativen Ausschlag in der EW-Komponente für das zweite Instrument.

Bei den anderen Stationen wurde für die Komponenten jeweils nur eine mathematische Funktion verwendet, um die Reduktion durchzuführen, bei der Station Stockau aber zwei: für die EW-Komponente der Logarithmus mit dem jeweils selben Zeitversatz von 500 Stunden und für die NS-Komponente eine lineare Funktion. Ein Grund dafür kann in der Transformation (Kapitel 6.2.4) der Kanäle auf NS-EW liegen. Zur Transformation wird der Azimut verwendet, welcher während der Installation ermittelt wurde. Eine falsche Azimut-Transformation wäre in den Gezeitenanalysen (Abb. 8.4 und 8.5) festgestellt worden. Durch den gleichen passenden Zeitversatz in der EW-Komponente kann für die allgemeine Drift die Geologie bzw. das regionale Stressfeld herangezogen werden. Sind die obigen Ausführungen zutreffend, so bleibt nur ein in-

strumenteller Einfluss übrig, wobei dann wieder das Problem zu den Driften der anderen Stationen besteht, welche mit geologischen Strukturen zusammenfallen (siehe folgende Absätze). Zu dem instrumentellen Effekt ist zu sagen, dass bei der Installation, bedingt durch Temperaturunterschiede und Spannungen zwischen Instrument und der Verrohrung, logarithmisches Einlaufen beobachtet werden kann. Ähnlichkeiten zu diesem Effekt sind in der EW-Komponente in den ersten Wochen der Registrierung zu sehen.

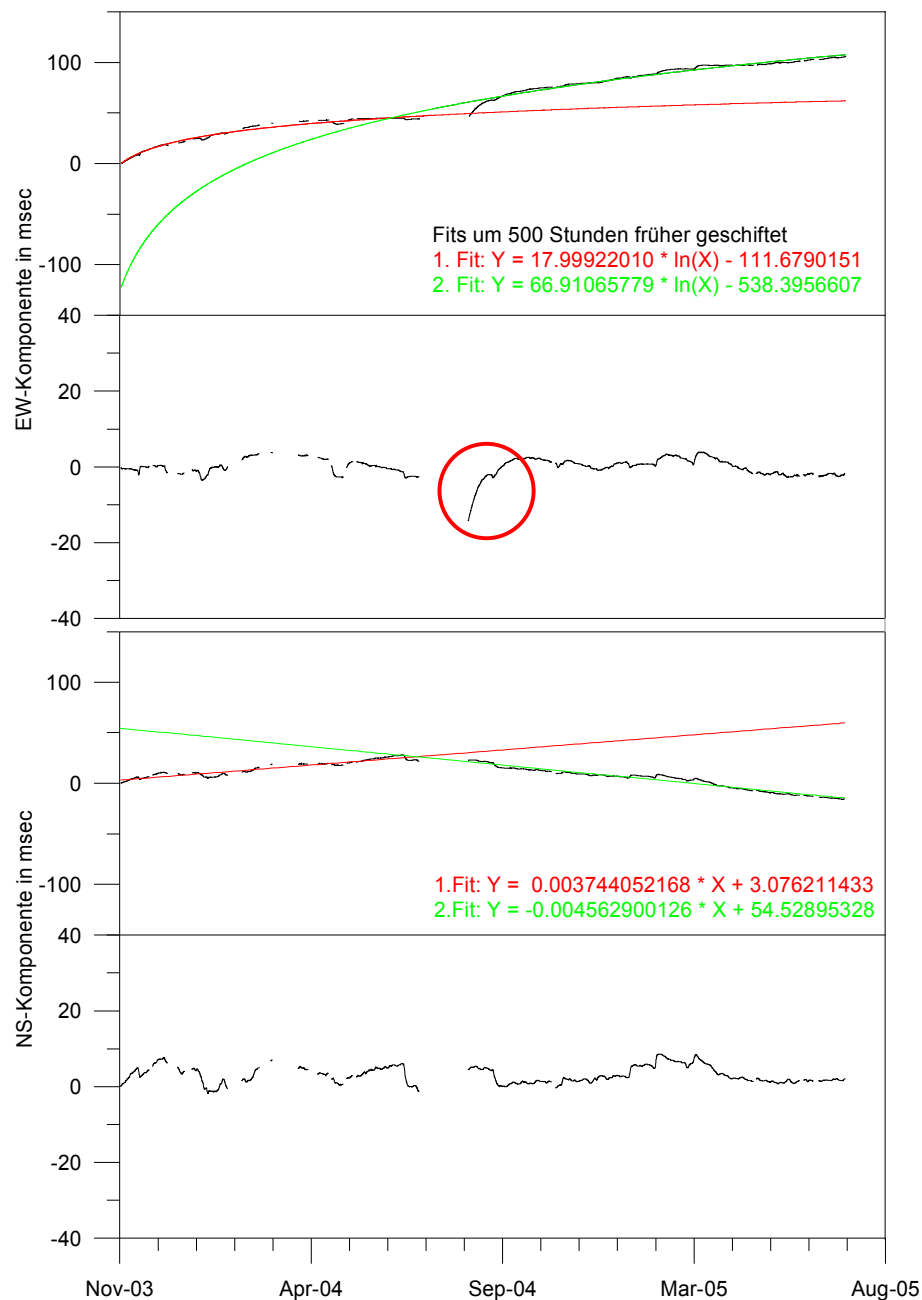


Abb. 9.5: Reduktion der Driften in den langperiodischen Anteilen der Neigung der Station Stockau.

Driften und Hodogramme

Für die Analyse und Interpretation der Hodogramme (Kapitel 8.4) müssen zusätzliche Informationen herangezogen werden. Aus den schon genannten Gründen wird die Diskussion im Folgenden nur auf die Daten der Stationen Berg, Eiglasdorf und Mittelberg beschränkt. In den Hodogrammen (Abb. 9.6) herrscht eine generelle NNW-SSE Richtung, welche mit der Richtung des regionalen Stressfeldes korreliert. Unter der Annahme, dass der Stress für diese Richtung verantwortlich ist, lässt sich dieser Anteil als Drift 1. Ordnung bezeichnen. Zu dieser lässt sich eine 2. Ordnung definieren, welche sich aus allen bekannten natürlichen Einflüssen (Luftdruck, Grundwasser, Temperatur, ...) außer der Geologie zusammensetzt. Die Geologie als Drift 3. Ordnung ist generell sehr stark mit dem Stressfeld verbunden, soll hier aber gesondert in der lokalen Umgebung behandelt werden. Die Richtungen des Westrandes des Falkenberg-Granits und der Fränkischen Linie fallen mit der des regionalen Stressfeldes zusammen. Daneben gibt es noch, nähert man sich der KTB, lokale Richtungen: die Erbdorfer-Linie, die Fichtelnaab-Störung und lokale Störungen.

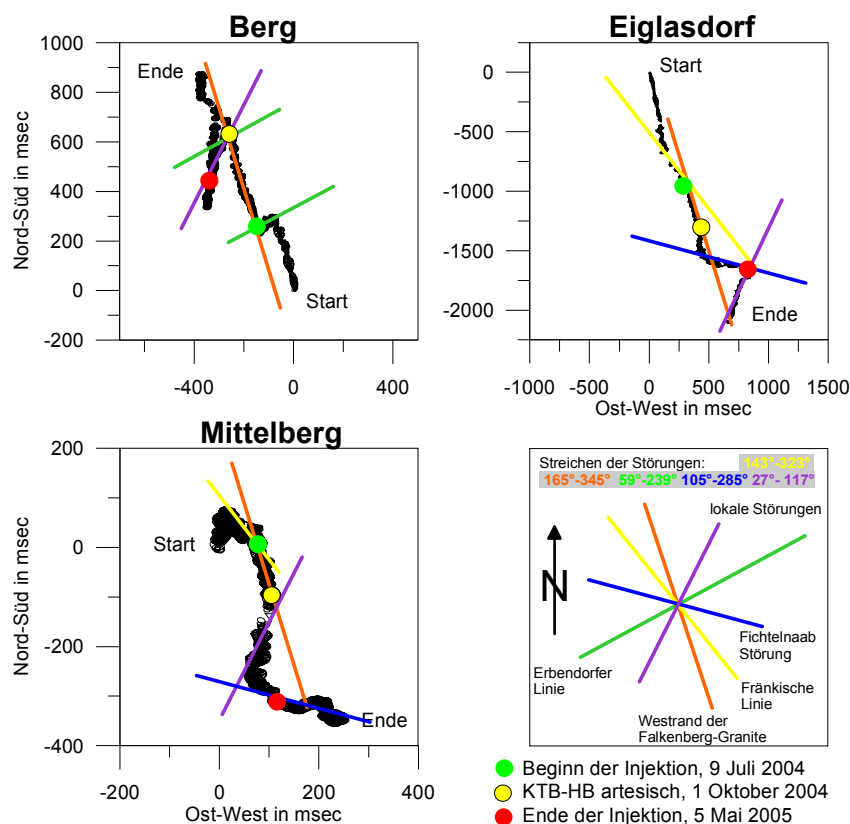


Abb. 9.6: Hodogramme der Stundendaten (Abb. 8.2) für die Stationen Berg, Eiglasdorf und Mittelberg.

Verallgemeinernd lässt sich unter dem Gesichtspunkt der Injektion sagen, dass die regionalen Richtungen durch lokale überlagert werden. Unter der Annahme der Richtigkeit dieser These müssten sich mit dem Abklingen der Druckerhöhung die regionalen Richtungen wieder einstellen, wie sich dies in Berg schon andeutet. Zur Untermauerung dieser These wären allerdings längere Zeitreihen nötig.

Nach Abzug der Drift 1. Ordnung (Abb. 8.16), also des Einflusses der regionalen Geologie und des Stressfeldes, müssten nur die Einflüsse aus der Drift 2. und 3. Ordnung in den Daten enthalten sein. Die Größenordnung der fassbaren einzelnen Elemente der Drift 2. Ordnung wurde bereits abgeschätzt und ist als kleiner Effekt (Abb. 8.16) noch in den Daten enthalten. Demnach bleibt als Restsignal nur noch die Drift, welche durch die Injektion ausgelöst wurde, übrig. Selbst der Versuch zur Reduktion des Grundwassers (Abb. 8.17) änderte die Fakten kaum. Der Grund dafür liegt aber wiederum im fast untrennbaren Zusammenhang der Injektion mit den Grundwasserspiegeln und den Neigungen. Um eine Abschätzung treffen zu können, ist eine Modellierung des Anstieges des Grundwassers für die betrachteten Stationen nötig. Nur so lässt sich sagen, ob in den etwa 20 msec Drift in radialer Richtung zur KTB integrativ die Einflüsse der regionalen Geologie, des Grundwassers und der erwarteten Neigung enthalten sind, welche den Effekt, falls dieser durch die Injektion ausgelöst wurde, verstärken können. Dies kann im Fall Eiglasdorf zum Beispiel durch die zur Fichtelnaab-Störung senkrecht stehenden Restdriftrichtung sein. Es ist nicht auszuschließen, dass der Effekt der passenden Drift nur scheinbar vorhanden ist und durch einen Einfluss möglicher saisonaler Faktoren und der Geologie überlagert wird.

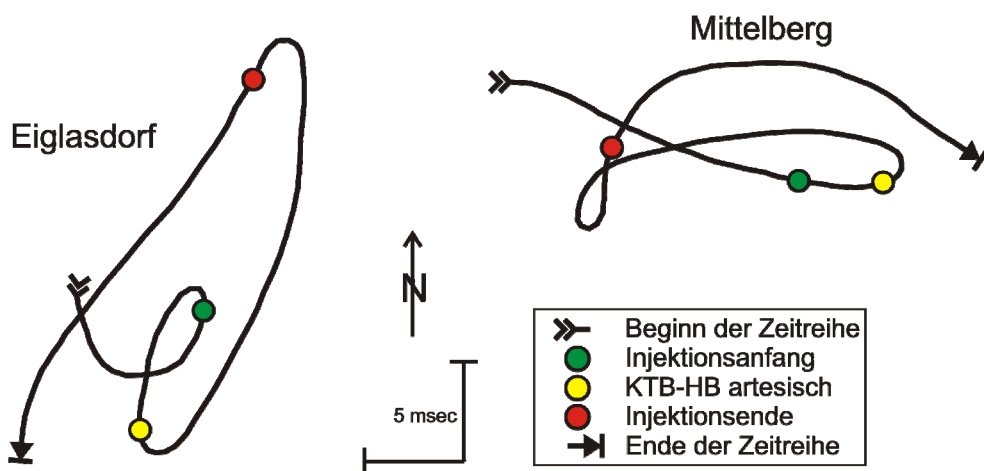


Abb. 9.7: Skizzierte Hodogramme der Stationen Eiglasdorf und Mittelberg.

Deutlich sichtbar ist die radiale Vorzugsrichtung der Drift (Abb. 9.7). Die zeitlichen Eckpunkte der Injektion sind durch Kreise markiert. Nach dem Injektionsbeginn sowie nach dem Zeitpunkt, als die Hauptbohrung artesisch wurde, tritt eine Driftrichtungsänderung um 180° ein. In Mittelberg tritt diese Inversion nur ein, nachdem die Hauptbohrung artesisch wurde. Der Grund dafür, dass keine Reaktion auf den Beginn der Injektion vorhanden ist, kann in der Geologie liegen. Unerklärlicherweise gibt es vor dem Ende der Injektion wiederum eine Änderung der Drift. Dieses eigenartige Phänomen wurde schon bei früheren Untersuchungen mit Pump und Injektionsversuchen festgestellt (Weise, 1992). Für die Injektion an der KTB bedeutet dies, dass das Ende der Injektion nicht mit der Änderung der Driftrichtung im direkten Zusammenhang steht.

Resultierende Schlussfolgerungen

In den obigen Ausführungen dieses Kapitels wurden Indizien gesammelt, um die eingangs formulierte Frage der Nachweisbarkeit der Injektion in den Zeitreihen zu beantworten. Bedingt durch die hohe Anzahl der Einflussfaktoren und des sehr kleinen erwarteten Signals, wodurch man schnell an die Grenze der Interpretierbarkeit der Daten gelangt, lässt sich dennoch sagen, dass durch die drei Indizien, den Grundwasserpegeln, den Parametervariationen und den Hodogrammen, die Injektion Effekte erzeugt hat, die messbar waren.

Für eine vollständige Interpretation jedoch sind, wie schon mehrfach angesprochen, Modellierungen nötig, in denen die gesammelten Zeitreihen und Fakten eingehen bzw. als Signal erklärt werden können. Zudem besteht das Problem, dass die Zeitreihen zu kurz sind.

Als Ergebnis könnte man umgekehrt formulieren: Wäre von der Injektion nur die Dauer bekannt, könnte man diese in den Neigungs- und Grundwasserdaten über den in dieser Arbeit gezeigten Weg nachweisen.

10 Zusammenfassung

Das Projekt, die Deformation der Kruste durch Fluidinjektion mittels eines Neigungsmesser-Arrays zu registrieren, bietet eine einzigartige Möglichkeit der Datenauswertung und Interpretation. Doch zuvor ist es notwendig die Zeitreihen aufzubereiten. Bedingt durch die große Menge an Daten der fünf ASKANIA-Bohrlochneigungsmesser war sehr schnell klar, dass das bisherige System zur Aufbereitung unzureichend war. Deshalb wurde dieses intensiv weiterentwickelt. Dabei sind die einzelnen neu- bzw. überarbeiteten Programme nicht nur für die Bearbeitung von Neigungsmesser-Zeitreihen geeignet, sondern bei allen Zeitreihen mit konstanter Samplerate.

In der Auswertung und Diskussion der Daten hat sich gezeigt, dass unter den getroffenen Annahmen und Einschränkungen die Auswirkungen der Injektion nachweisbar, in ihrer Größenordnung aber nicht abschätzbar sind. Mit Modellierungen wären die Größenordnungen der Effekte quantifizierbar.

Danksagung

Jeder, der in der Wissenschaft tätig ist, kennt die Mühen und Wehen einer Abschlussarbeit. Jeder hilfreiche Rat ist während dieser Zeit mehr als willkommen. Die wichtigste Unterstützung sollte dabei immer von den betreuenden Wissenschaftlern kommen, was nicht bei jedem Akademiker der Fall ist. Ich kann von Glück sprechen, dass Prof. Gerhard Jentzsch, Dr. Thomas Jahr und Dr. Horst Letz mich auf dem letzten Stück Weg meiner studentischen Laufbahn begleitet haben. Ihre konstruktive Kritik und ihre Motivation halfen mir, in der Datenflut immer den Kopf über Wasser zu halten und nicht im Datenmeer zu ertrinken. Gerade ihre Anleitung ist die größte Hilfe während des Schreibens meiner Diplomarbeit gewesen.

Neben diesen drei Wissenschaftlern waren es auch Prof. Corinna Kroner, Dr. Adelheid Weise und Prof. Hans-Joachim Kümpel, die mir das eine oder andere Mal den Impuls in die richtige Richtung zur Lösung eines Problems gaben, ohne dabei ein geistiges Erdbeben auszulösen, das meine Diplomarbeit in sich hätte zusammenstürzen lassen. Auch ihnen sei ein herzlicher Dank gesagt.

An dieser Stelle möchte ich auch allen Mitarbeitern im Arbeitskreis KTB danken, ohne deren Datensammlung diese Arbeit nicht zustande gekommen wäre.

Zum Schluss noch ein Dank an die Leseratten, die sich durch die manchmal sehr naturwissenschaftliche Sprache meiner Arbeit hangelten und hier und da ein Wörtchen oder Komma einfügten, das eine oder andere entfernten und das Dickicht der deutschen Rechtschreibung ein wenig lichteten.

Fürchte dich nicht langsam zu gehen, sondern fürchte dich stehen zu bleiben.

Chinesisches Sprichwort

Literatur

- Agnew, D. C. (1986): Strainmeters and Tiltmeters, *Rev. Geophys.*, 24/3, 579-624.
- Asch, G. (1988): Die Registrierung langperiodischer Signale mit geophysikalischen Sensoren hoher Dynamik. Dissertation, Freie Univ. Berlin, Berliner Geowiss. Abh., Reihe B, Bd.15, Reimer Verlag Berlin.
- Asch, G., Kind, R., Rabbel, W., Shapiro, S. A. (2003): Seismological Monitoring of the KTB Injektion Project, Continuation Proposed (GFZ, FUB, unveröffentlicht).
- Askania (1968): Bohrloch-Gezeitenpendel Gbp1 – Betriebsanleitung, Askania-Werke, Berlin (unveröffentlicht).
- Bartels, J. (1957): Gezeitenkräfte, *Handbuch der Physik* (Hrsg. Flügge, S.), Band XLVIII, Geophysik 2, 734-774.
- Biot, M. A. (1941): General theory of three-dimensional consolidation. *J. Appl. Geoph.*, 12, 155-164.
- Chojnicki, T. (1973): Ein Verfahren zur Erdgezeitenanalyse in Anlehnung an das Prinzip der kleinsten Quadrate. *Mitt. Inst. Theor. Geodäsie*, 15, Univ. Bonn.
- Dehant, V. (1987): Tidal parameters for an inelastic Earth. *Physics of the Earth and Planetary Interiors*, 49, 97-116.
- De Toro, C., Venedikov, A. P. and R. Vieira (1993): A new method of earth tide data analysis. *Bull. d'Inf. Marées Terrestres*, 116, 8557-8586.
- Doodson, A. T. (1922): The harmonic development of the tide generating potential. *Proc. Royal Soc. London, Series A* 100, 306-328.
- Fabian, M. (2004): Near Surface Tilt und Pore Pressure Changes Induced by Pumping in Multi-Layered Poroelastic Half-Spaces. Dissertation. Berichte, Fachbereich Geowissenschaften, Friedrichs-Wilhelm-Universität Bonn, No. 229, S. 121.
- Flach, D., Große-Brauckmann, W. (1974): Recordings with the Askania borehole tiltmeter (vertical pendulum) in the frequency range of earth's free oscillations, *J. Geophys.* 40, 355-365.
- Gomez, M., Jentzsch, G., Ramatschi, M. and D. Flach (1995): ASKANIA-Borehole-Tiltmeters: Test of nine different instruments regarding the orthogonality of both channels. *Bull. d'Inf. Marées Terrestres*, 121, 9076-9082.
- Graf, A. (1958): Bemerkungen zur instrumentellen Ausrüstung einer Gezeitenstation. – *Comm. Obs. Roy. Belg.* 142, Ser. Geophys. 47, 116-122.

- Graf, A. (1964): Erste Neigungsmessungen mit dem Vertikalpendel in einem 30 m Bohrloch. Comm. Obs. Roy. Bel., 236, Ser. Geophys. Nr. 69, 5ieme Symp. Int. sur la marées terrestres, Brüssel, 249-254.
- Gräsele, W., Kessels, W., Kümpel, H.-J., Li, Xuan (2006): Hydraulic observations from a one year fluid production test in the 4000 m deep KTB pilot borehole. – *Geofluids*, 6, 8-23.
- Graupner, S. (2001): Hochpräzise Neigungsmessungen im geophysikalischen Observatorium Nokogiriyama/Tokio. Diplomarbeit. Institut für Geowissenschaften. FSU Jena, 191 S.
- Große-Brauckmann, W. (1979): Untersuchung der Bodenunruhe im Periodenbereich von 2 Minuten bis 1 Stunde mit dem ASKANIA-Bohrlochneigungsmesser. Dissertation, Techn. Universität Clausthal, 130 S.
- Große-Brauckmann, W. and O. Rosenbach (1979): Resolution and Stability of Borehole Tiltmeters. In: A. Vogel (Ed.): *Terrestrial and space techniques in earthquake prediction research*. Vieweg & Sohn, Braunschweig, 712 S.
- Harrison, J.C. (1976): Cavity and topographic effects in tilt and strain measurements, *JGR*, 81/2, 319-328.
- Harrison, J.C., Herbst, K. (1976): Thermoelastic strains and tilts revisited. *Geophysical Research Letters*. 4/11.
- Herbst, K. (1976): Interpretation von Neigungsmessungen im Periodenbereich oberhalb der Gezeiten. Dissertation, Techn. Univ. Clausthal, 180 S.
- Hirschmann, G., Lapp, M. (1995): Evaluation of the structural geology of the KTB Hauptbohrung. KTB-Report 94-1. Niedersächsisches Landesamt für Bodenforschung. Hannover.
- Hirschmann, G. (1996): KTB – The structure of a Variscan terrane boundary: seismic investigation – drilling – models. – *Tectonophysics* 264: 327-339.
- Ishii, H., Jentzsch, G., Graupner, S., Nakao, S., Ramatschi, M. and Weise, A., 2001. Observatory Nokogiriyama / Japan: Comparison of different tiltmeters. *Proc. 14th International Symposium on Earth Tides. Special Issue of the Journal of the Geodetic Soc. of Japan*, 47/1, 155 – 160.
- Jacoby, H. D. (1966): Das neue Bohrloch-Gezeitenpendel nach Graf. – *Askania-Warte*, 67, 12-17.
- Jahr, T., Jentzsch, G., Letz, H., Sauter, M. (2005): Fluid injection and surface deformation at the KTB location: modelling of expected tilt effects. *Geofluids* 5, 20-27.
- Jahr, T., Letz, H. & Jentzsch, G. (2006): The ASKANIA borehole tiltmeter array at the KTB location / Germany. – *Journ. of Geodyn.*, 41, 190-197.
- Jentzsch, G. (1976): Separation von Erdgezeitenwellen durch numerische Filterung und Regressionsanalyse Dissertation, Techn. Universität Clausthal 89S.
- Jentzsch, G., Liebing, M., Weise, A. (1993): Deep boreholes for high resolution tilt recordings, *Bulletin d'Inf. Marées Terrestres*, 115, 8488-8509.
- Jentzsch, G., and S. Koß (1997): Interpretation of long-period tilt records at Blå Sjø, Southern Norway, with respect to the variations of the lake level. *Phys. Chem Earth*, 22, 25 - 31.

- Jentzsch, G., Jahr, T., Letz, H., Gebauer, A. (2005): Die Erdeigenschwingungen nach dem Sumatra-Erdbeben vom 26.12.2004 - Beobachtet mit fünf ASKANIA-Bohrloch-Neigungsmessern an der KTB, DDG Mittlg 1/2005, S. 12, 13.
- Johnson, H. O., Wyatt, F., Agnew, D. C. and W. Zürn (1995): Tidal tilts at Pinyon Flat, California measured at depths of 24 and 120 meters. Proc. 12th Int. Symp. Earth Tides, H.T. Hsu (Ed.), Science Press, Beijing, 1995, 129-135.
- Kaiser, J. (1950): Untersuchungen über das Auftreten von Geräuschen beim Zugversuch, Dissertation, Technische Hochschule München, 38 S.
- Kossmat, F. (1927): Gliederung des variskischen Gebirgsbaues. – Abh. Sächs. Geol. L.-Amt, 1: 1-39, Leipzig.
- Kümpel, H.-J. (1982): Tilt measurements. What do they tell us? Terra cognita, 2, 391-399.
- Kümpel, H.-J. (1988): Gesteinsverformungen durch Porendruckgefälle: Neigungsmessungen in der Umgebung von Brunnen und Interpretation für homogene, poröse, elastische Medien, Kurzber. aus d. Bauforschung, Ber. Nr. 39, 145-146.
- Kümpel, H.-J. (1989): Verformungen in der Umgebung von Brunnen. Habilitationsschrift für das Fach Geophysik an der Mathematisch-Naturwissenschaftlichen Fakultät der Christian-Albrechts-Universität Kiel, 198 S.
- Kümpel, H.-J., Varga, P., Lehmann, K., Mentés, Gy. (1996): Ground tilt induced by pumping – preliminary results from the Nagycenk test site Hungary. Acta Geod. Geophys Hung., 31, No. 1 – 2, 67 - 78.
- Kümpel, H.-J. (2005): Vorlesungsscript Poroelastische Theorie, http://www.ifg.tu-clausthal.de/Poro_komplett.pdf
- Lehmann, K. (2001): Porendruckinduzierte Neigungssignale in geringen Tiefen und ihre Modellierung im homogenen Halbraum, Dissertation, Friedrichs-Wilhelm-Universität Bonn, Shaker Verlag, S. 161.
- Melchior, P., (1983): The tides of the planet earth, 2nd Edition. Pergamon Press, Oxford, 641 S.
- Pertsev, B. (1959): Ob ouschetie spolzaniya nulia pir nabloudenij ouprougikh prilivov, Izv. Akad. Nauk SSR, No. 4.
- Ramatschi, M. (1994): Funktionstest und Kalibrierung der ASKANIA-Bohrloch-neigungsmesser. (Unveröffentlichter Prüfbericht).
- Schneider, M. (1965): Messung der Lotschwankungen mit Vertikalpendeln in Freiberg/Sa. – DGK, Reihe C, Nr. 79.
- Schüller, K. (1986): Simultaneous tidal and multi-channel input analysis as implemented in the HYCON-Method. Proc. 10th Int. Symp. Earth Tides, R. Vieira (Ed.), Madrid, 1986, 515-520.
- Simon, D., Schneider, M. (1973): Analyse der nichtperiodischen Bodendeformation an der Erdgezeitenstation Tiefenort 1958-1973. Geodät. Geophys. Veröff. Reihe III Sitzungsprotokoll, KTB-Treffen vom 15.05.2003 (unveröffentlicht).
- Stettner, G. (1992): Geologie im Umfeld der Kontinentalen Tiefbohrung Oberpfalz, Bayrisches Geologisches Landesamt, München, 240 S.
- Tamura, Y., Sato, T., Ooe, T. and M. Ishigiro (1991): A procedure for tidal analysis with a Bayesian information criterion. Geophys. J. Int., 104, 507-516.

- Torge, W. (1989): Gravimetry, Berlin – New York, 465 S.
- Venedikov, A. P. (1966): Une méthode pour l'analyse des marées terrestres à partir d'enregistrement de longueur arbitraire. Comm. Observ. Royal Belgique, 71.
- Vogel, A. (1979): Tilt measurement techniques in earthquake research. In: A. Vogel (Ed.): Terrestrial and space techniques in earthquake prediction research. Vieweg & Sohn, Braunschweig, 712 S.
- Wang, H. F. (2000): Theory of Linear Poroelasticity – with Application to Geomechanics and Hydrology, Princeton University Press, Princeton, 287 S.
- Weise, A. (1992): Neigungsmessungen in der Geodynamik - Ergebnisse von der 3-Komponentenstation Metsähovi. Dissertation, Techn. Univ. Clausthal, 180 S.
- Weise, A., Jentzsch, G., Kiviniemi, Kääriäinen (1999): Comparison of longperiod tilt measurements: results from two clinometric stations Metsähovi and Lohja, Finland. Journal of Geodynamics, 27, 237-257.
- Wenzel, H.-G. (1996): The Nanogal Software: Earth Tide Data Processing Package ETERNA 3.30., Bull. d'Inf. Marées Terrestres 124, 9425-9439.
- Wenzel, H.-G. (1997a): Tide-Generating Potential for the Earth. In: Wilhelm, H., Zürn, W. and H.-G. Wenzel (Eds.): Tidal Phenomena. Lecture Notes in Earth Sciences, 66, Springer, Berlin, 398 S.
- Wenzel, H.-G. (1997b): Analysis of Earth Tide Observations. In: Wilhelm, H., Zürn, W. and H.-G. Wenzel (Eds.): Tidal Phenomena. Lecture Notes in Earth Sciences, 66, Springer, Berlin, 398 S.
- Westerhaus, M., and J. Zschau, (2001). No clear evidence for temporal variations of tidal tilt prior to the 1999 Izmit and Düzce earthquakes in NW-Anatolia. Proc. 14th International Symposium on Earth Tides, Special Issue of the Journal of the Geodetic Soc. of Japan, 47/1, 448– 455.
- Widmer, R., W. Zürn & G. Masters (1992): Observation of low-order toroidal modes from the 1989 Macquarie Rise event. Geophys. J. Int., 111/2, 226 – 236.
- Zschau, J. (1974): Lotschwankungsanomalien in Erdgezeiten-Registrierungen mit dem ASKANIA-Bohrloch-Vertikal-Pendel nach Graf. Dissertation, Univ. Kiel, 214 S.
- Zürn, W. & R. Widmer-Schmidrig (2002): Globale Eigenschwingungen der Erde. Physik Journal, 1/10.

Anhang

I Liste der Symbole

Symbol	Beschreibung	Einheit
\bar{g}	Schwerevektor	$\frac{m}{s^2}$
M	Masse	kg
γ	Gravitationskonstante	$\frac{m^3}{kgs^2}$
\bar{r}, \bar{s}	Abstandsvektoren	m
V	Gravitationspotential	$\frac{m^2}{s^2}$
G	Gezeitenkonstante	$\frac{m^2}{s^2}$
e_{ij}	Komponente des Verformungstensors	-
σ_{ij}	Komponente des Spannungstensors	Pa
E	Elastizitätsmodul (drainierte Bedingungen)	Pa
E_u	Elastizitätsmodul (undrainierte Bedingungen)	Pa
P	Porendruck	Pa
P_c	Umgebungsdruck	Pa
P_u	Druckänderung	Pa
u_i	Komponenten des Verschiebungsvektors	m
x_i, x_j	Raumkoordinaten	m
α	Koeffizient des effektiven Drucks	-

μ	Schermodul	Pa
Q	Fluidinjektionsrate	$\frac{m^3}{kg}$
V_f	Fluidvolumen	m^3
m_f	Fluidmasse	kg
ρ_f	Fluiddichte	$\frac{kg}{m^3}$
ν	Poisson-Verhältnis (drainierte Bedingungen)	-
ν_u	Poisson-Verhältnis (undrainierte Bedingungen)	-
B	Skempton's Porendruckparameter	-
D	hydraulische Diffusivität	$\frac{m^2}{s}$
κ	Permiabilitätsmaas	$\frac{m^3 s}{kg}$
M^*	Substitutionsterm	-
\mathfrak{S}	Substitutionsterm	-
R	Abstand zur Punktquelle	m
\mathfrak{R}	dimensionsloser Abstand zur Punktquelle	-
ϵ, ϵ^-	Errorfunktionen	-
t	Zeit	s
t^*	dimensionslose Zeit	-
z	Vertikalabstand zur Punktquelle (Zylinderkoordinaten)	m
r	Horizontalabstand zur Punktquelle (Zylinderkoordinaten)	m
θ	Winkelkoordinate (Zylinderkoordinaten)	$^\circ$
Δ_v	Dilatation	-
Δ_r	Radialverformung	-
Δ_T	Tangentialverformung	-
Δ_S	Scherverformung e_{rz}	-
δ_{ij}	Kronecker-Symbol	-
γ_R, γ_w	Gezeitenparameter	-

II Programmcode

II.I Programmcode - *DatAdd.exe*

Der Programmcode von *DatAdd.exe* ist folgend dargestellt. Danach ist das Script der Batch-Datei *DatAdd.bat* zu finden, mit welcher das Programm *DatAdd.exe* gesteuert werden kann.

```
using System;
using System.IO;
using System.Globalization;

namespace DatAdd
{
    class Add
    {
        #region Globale Variablen
        private static string[] sourcefile = null;
        private static string destinationfile = null;
        private static string reportfile = null;

        private static FileStream sourcestream = null;
        private static StreamReader sourcereader = null;
        private static FileStream destinationstream = null;
        private static StreamWriter destinationwriter = null;
        private static FileStream reportstream = null;
        private static StreamWriter reportwriter = null;

        private static DateTime StartTime = new DateTime();
        private static DateTime Time = new DateTime(1,1,1,1,1,1);
        private static TimeSpan timeSpan = new TimeSpan(10*10000000);

        private static string LueckeAlt = "0";
        private static string LueckeNeu = "99999.999";

        private static long anz = 0;
        private static long all = 0;
        private static long l = 0;
        private static long ll = 0;
        // Zahlen sind durch Dezimalpunkt getrennt
        public static NumberFormatInfo NumberFormat = new CultureInfo( "en-US",
            false).NumberFormat;
        #endregion

        [STAThread]
        static void Main(string[] args)
        {
            //erster Check der Eingabeparameter
            if (args.Length < 3)
            {
                showusage();
                return;
            }
            // Initialisierung und Kontrolle der Parameter
            string station = args[0];
            string kanal = args[1];
            if ( station.Length != 1 || kanal.Length != 1)
            {
                showusage();
                return;
            }
            string start = args[2];
            if ( start.Length != 5)
            {
```

```
        showusage();
        return;
    }

    string end          = null;
    if (args.Length < 4 && end.Length !=5)
    {
        showusage();
        return;
    }

    // Verfahren für ein oder mehrere umzuwandelnde Dateien um Zieldateinamen zu
    // erstellen
    if (args.Length < 4)
    { // Für eine Rohdatei
        end = start;
        destinationfile = Environment.CurrentDirectory + "\\\" +
            station.ToUpper()+kanal.ToLower()+start+".dat";
        reportfile = Environment.CurrentDirectory + "\\\" +
            station.ToUpper()+kanal.ToLower()+start+".Add";
    }
    else
    { // Für mehrere Rohdateien
        end = args[3];
        destinationfile = Environment.CurrentDirectory + "\\\" +
            station.ToUpper()+kanal.ToLower()+start+"-"+end+".dat";
        reportfile = Environment.CurrentDirectory + "\\\" +
            station.ToUpper()+kanal.ToLower()+start+"-"+end+".Add";
    }
    //Konsolen Meldung
    string space = null;
    for (int i = 0 ; i < 32 - Path.GetFileName(destinationfile).Length;i++)
        space += " ";
    Console.WriteLine();
    Console.WriteLine("*****");

    Console.WriteLine("* Dateien zus.-fuegen für: "
        +Path.GetFileName(destinationfile)+space+"(c) A.G. Ver.: 1.0 *");
    Console.WriteLine("*****");

    Console.WriteLine();

    // weitere Initialisierung der Parameter
    TimeSpan de = new DateTime(Convert.ToInt32("20"+end.Substring(0,2)),1,
        1).AddDays(Convert.ToSingle(end.Substring(2,3))-1)
        - new DateTime(Convert.ToInt32("20"+start.Substring(0,2)),1,
        1).AddDays(Convert.ToSingle(start.Substring(2,3))-1);
    int delta = (int)(de.TotalDays);

    if (kanal.ToLower() == "x")
        kanal = "VAE";
    if (kanal.ToLower() == "y")
        kanal = "VAN";

    // Ermitteln der notwendigen Dateien
    Console.WriteLine("Dateinamen für Umformatierung werden ermittelt!\n");
    if(!getFileName(station, kanal, start, delta))
    { // Ausgabe falls anfangs oder Enddatei nicht vorhanden sind
        Console.WriteLine("\nStart- oder Enddatei existieren nicht!\n");
        return;
    }
    // Öffnen der ersten Quelldatei
    sourcestream = new FileStream(sourcefile[0], FileMode.Open);
    sourcereader = new StreamReader(sourcestream);
    // Notwendiger string für Report
    string sourceFile = Path.GetFileName(sourcefile[0]);
    // Startzeit der enthaltenen Daten ermitteln
    getStartTime();

    // Zieldatei und Reportdatei erstellen und öffnen
    open();
    // Header für Zieldatei erzeugen
    MakeHeader();

    // Zeitkennung erzeugen und Werte umformatieren und kopieren
```

```

for (int j = 1 ; j <= delta; j++)
{
    // Zeitcheck und Verfahren zwischen den Dateien
    if(j<=delta)
        timeBetwenCheck();
    // Zeitkennung erzeugen, Werte umformatieren und kopieren
    copyValue();
    sourcereader.Close();
    // Report
    Console.WriteLine("Datei: " + sourceFile + " enthielt " + anz + " Werte und "
        + 1 + " Luecken!");
    reportwriter.WriteLine("Datei: " + sourceFile + " enthielt " + anz +
        " Werte und " + 1 + " Luecken!");

    // Auf nächste Datei prüfen und/oder weitere suchen
    if(j<delta)
    {
        // folgende Datei prüfen
        while (sourcefile[j] == null )
        {
            // Reportmeldung
            DateTime st = new DateTime(Convert.ToInt32("20"+start.Substring(0,2))
                ,1,1).AddDays(Convert.ToSingle(start.Substring(2,3))-1+j);
            Console.WriteLine("Datei: " + station.ToUpper() + "***.KT."
                + kanal + ".D." + st.Year.ToString("#0000") + "." +
                st.DayOfYear.ToString("#000") + "*****.ASCII existiert nicht!");
            reportwriter.WriteLine("Datei: "+ station.ToUpper() + "***.KT."
                + kanal + ".D." + st.Year.ToString("#0000") + "." +
                st.DayOfYear.ToString("#000") + "*****.ASCII existiert nicht!");
            j++;
        }
        // folgende Datei öffnen
        if (sourcefile[j] != null)
        {
            sourceFile = Path.GetFileName(sourcefile[j]);
            sourcestream = new FileStream(sourcefile[j], FileMode.Open);
            sourcereader = new StreamReader(sourcestream);
            getStartTime();
        }
    }
    all += anz;
}
all += 11;
// Report
Console.WriteLine("\nDatei : " + Path.GetFileName(destinationfile)+" enthaelt "
    + all + " Werte und "+ 11 + " Werte ergaenzt!");
reportwriter.WriteLine("\nDatei : " + Path.GetFileName(destinationfile) +
    " enthaelt " + all + " Werte und "+ 11 + " Werte ergaenzt!");
Console.WriteLine("\nReportdatei: " + Path.GetFileName(reportfile));

// Alle Dateien schließen
close();
}

#region Funktionen
private static bool getFileName(string station,string kanal,string start,
    int delta)
{
    sourcefile = new string[delta];
    for (int i = 0; i < delta; i++)
    {
        string[] dirs = Directory.GetFiles(Environment.CurrentDirectory);
        try
        {
            foreach (string dir in dirs)
            {
                string s = Path.GetFileName(dir);

                DateTime st = new DateTime(Convert.ToInt32("20"+start.Substring(0,2))
                    ,1,1).AddDays(Convert.ToSingle(start.Substring(2,3))-1+i);
                int st1 = st.Year - 2000;

                if( s.Length == 32 && s.Substring(0,1) == station.ToUpper() &&

```

```
s.Substring(7,3) == kanal && s.Substring(15,2) ==
stl.ToString("#00") && s.Substring(18,3) ==
st.DayOfYear.ToString("#000") && s.Substring(27,5) == "ASCII")
sourcefile[i] = dir;
    }
}
catch
{
    sourcefile[i] = null;
}
}
if(sourcefile[0] == null || sourcefile[delta-1] == null)
    return false;
else
    return true;
}
private static void timeBetwenCheck()
{
    if (Time == new DateTime(1,1,1,1,1,1))
    {
        Time = StartTime;
    }
    else
    {
        TimeSpan diff = StartTime - Time;
        TimeSpan dNULL = new TimeSpan(0);

        if ( diff > dNULL)
        {
            while( Time != StartTime)
            {
                destinationwriter.WriteLine(Time.Year.ToString("#0000") +
                    Time.Month.ToString("#00")+Time.Day.ToString("#00") + " " +
                    Time.Hour.ToString("#00")+Time.Minute.ToString("#00")+Time.Second.T
                    oString("#00") + format(Convert.ToDouble(LueckeNeu,NumberFormat)));
                Time += TimeSpan;
                ll++;
            }
        }
        if ( diff < dNULL)
        {
            while( Time != StartTime)
            {
                sourcereader.ReadLine();
                StartTime += TimeSpan;
                all--;
            }
        }
    }
    ll = ll;
}
private static void copyValue()
{
    string str = null;
    double v = 0;
    anz = 0;
    l = 0;
    while ( (str = sourcereader.ReadLine()) != null && str.Length >=1)
    {
        v = Convert.ToDouble(str,NumberFormat)*1000;
        if ( v == Convert.ToDouble(LueckeAlt,NumberFormat) )
        {
            v = Convert.ToDouble(LueckeNeu,NumberFormat);
            l++;
        }
        destinationwriter.WriteLine(Time.Year.ToString("#0000") +
            Time.Month.ToString("#00") + Time.Day.ToString("#00") + " " +
            Time.Hour.ToString("#00") + Time.Minute.ToString("#00") +
            Time.Second.ToString("#00")+ format(v));
        Time += TimeSpan;
        anz++;
    }
}
private static void showusage()
```

```

{
    Console.WriteLine();
    Console.WriteLine("*****");
    Console.WriteLine("*****");
    Console.WriteLine("* Das Programm benötigt 3 Eingabe Argumente!");
    Console.WriteLine("*****");
    Console.WriteLine("*****");
    Console.WriteLine("* DatAdd [Station:('B','E','S','M','P')] [Kanal:('x','y')]");
    Console.WriteLine("*****");
    Console.WriteLine("*****");
    Console.WriteLine("Das Programm fuegt einzelne Tagesdateien zu einer langen");
    Console.WriteLine("Datenreihe im");
    Console.WriteLine("*****");
    Console.WriteLine("Eternaformat zusammen.");
    Console.WriteLine("*****");
    Console.WriteLine("Start und Ende haben das Format 'jjddd' (der Tag ist");
    Console.WriteLine("julianisch anzugeben");
    Console.WriteLine("*****");
    Console.WriteLine("3.0 André Gebauer");
    Console.WriteLine("*****");
    Console.WriteLine("*****");
    Console.WriteLine();
}
private static void open()
{
    destinationstream = new FileStream(destinationfile, FileMode.Create);
    destinationwriter = new StreamWriter(destinationstream);

    reportstream = new FileStream(reportfile, FileMode.Create);
    reportwriter = new StreamWriter(reportstream);
}
private static void close()
{
    sourcereader.Close();

    destinationwriter.Flush();
    destinationwriter.Close();

    reportwriter.Flush();
    reportwriter.Close();
}
private static DateTime ConvertToDate(string str)
{
    string tmp = str.Substring(17, str.Length-17);
    float second = Convert.ToSingle(tmp, NumberFormat);
    if(second >= 0 && second < 5) second = 0;
    if(second >= 5 && second < 15) second = 10;
    if(second >= 15 && second < 25) second = 20;
    if(second >= 25 && second < 35) second = 30;
    if(second >= 35 && second < 45) second = 40;
    if(second >= 45 && second < 55) second = 50;
    if(second >= 55 && second < 60) second = 0;

    return (new DateTime(Convert.ToInt32(str.Substring(0,4)),
        Convert.ToInt32(str.Substring(5,2)), Convert.ToInt32(str.Substring(8,2)),
        Convert.ToInt32(str.Substring(11,2)), Convert.ToInt32(str.Substring(14,2)), (int)(second)));
}
private static void getStartTime()
{
    for(int i= 0;i<3;i++)
        sourcereader.ReadLine();
    string str = sourcereader.ReadLine();
    StartTime = ConvertToDate(str);
    StartTime = StartTime;
    for(int i= 0;i<4;i++)
        sourcereader.ReadLine();
}
private static void MakeHeader()
{
    destinationwriter.WriteLine("File :Pendel XY-Kanal");
    destinationwriter.WriteLine("Status :Askania KTB");
    destinationwriter.WriteLine("Start : "+StartTime.Year.ToString("#0000")+ "

```

```

        +StartTime.Month.ToString("#00")+" " +StartTime.Day.ToString("#00")+" "
        +StartTime.Hour.ToString("#00")+" " +StartTime.Minute.ToString("#00")+" "
        +StartTime.Second.ToString("#00"));
    destinationwriter.WriteLine("End          :");
    destinationwriter.WriteLine("Comment      :");
    destinationwriter.WriteLine("");
    destinationwriter.WriteLine("");
    destinationwriter.WriteLine("");
    destinationwriter.WriteLine("Remarks    :");
    destinationwriter.WriteLine("");
    destinationwriter.WriteLine("C*****");
    destinationwriter.WriteLine("ASK          1.000");
    destinationwriter.WriteLine("77777777    .000");
}
private static string format(double v)
{
    string tmpvalue = v.ToString("0.000", NumberFormat);
    string tmpspace =null;
    for (short x=0;x < 10-tmpvalue.Length;x++)
        tmpspace += " ";
    return tmpspace+tmpvalue;
}

#endregion
}
}

```

II.II Script der Batch-Datei *DatAdd.bat*

```

echo off

if "%1" == "" GOTO 1
if "%2" == "" GOTO 1
if "%3" == "" GOTO 1

if "%1" == "ALL" GOTO ALL
if "%1" == "All" GOTO ALL
if "%1" == "all" GOTO ALL

if "%1" == "B" GOTO B
if "%1" == "b" GOTO B
if "%1" == "E" GOTO E
if "%1" == "e" GOTO E
if "%1" == "S" GOTO S
if "%1" == "s" GOTO S
if "%1" == "M" GOTO M
if "%1" == "m" GOTO M
if "%1" == "P" GOTO P
if "%1" == "p" GOTO P

if "%1" == "Bx" GOTO Bx
if "%1" == "bx" GOTO Bx
if "%1" == "Ex" GOTO Ex
if "%1" == "ex" GOTO Ex
if "%1" == "Sx" GOTO Sx
if "%1" == "sx" GOTO Sx
if "%1" == "Mx" GOTO Mx
if "%1" == "mx" GOTO Mx
if "%1" == "Px" GOTO Px
if "%1" == "px" GOTO Px

if "%1" == "By" GOTO By
if "%1" == "by" GOTO By
if "%1" == "Ey" GOTO Ey
if "%1" == "ey" GOTO Ey
if "%1" == "Sy" GOTO Sy
if "%1" == "sy" GOTO Sy
if "%1" == "My" GOTO My
if "%1" == "my" GOTO My
if "%1" == "Py" GOTO Py

```



```
if "%1" == "py" GOTO Py

GOTO 1

rem berg*****
:B
cd DatenRoh\1_Berg_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\1_Berg_X\*.dat Daten\1_Berg
move DatenRoh\1_Berg_X\*.Add Daten\1_Berg

cd DatenRoh\1_Berg_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\1_Berg_Y\*.dat Daten\1_Berg
move DatenRoh\1_Berg_Y\*.Add Daten\1_Berg
GOTO 2

:Bx
cd DatenRoh\1_Berg_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\1_Berg_X\*.dat Daten\1_Berg
move DatenRoh\1_Berg_X\*.Add Daten\1_Berg
GOTO 2

:By
cd DatenRoh\1_Berg_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\1_Berg_Y\*.dat Daten\1_Berg
move DatenRoh\1_Berg_Y\*.Add Daten\1_Berg
GOTO 2

rem eig*****
:E
cd DatenRoh\2_Eiglasdorf_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_X\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_X\*.Add Daten\2_Eiglasdorf
cd DatenRoh\2_Eiglasdorf_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_Y\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_Y\*.Add Daten\2_Eiglasdorf

GOTO 2

:Ex
cd DatenRoh\2_Eiglasdorf_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_X\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_X\*.Add Daten\2_Eiglasdorf
GOTO 2

:Ey
cd DatenRoh\2_Eiglasdorf_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_Y\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_Y\*.Add Daten\2_Eiglasdorf
GOTO 2

rem sto*****
:S
cd DatenRoh\3_Stockau_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\3_Stockau_X\*.dat Daten\3_Stockau
move DatenRoh\3_Stockau_X\*.Add Daten\3_Stockau
cd DatenRoh\3_Stockau_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\3_Stockau_Y\*.dat Daten\3_Stockau
```

```
move DatenRoh\3_Stockau_Y\*.Add Daten\3_Stockau
GOTO 2

:Sx
cd DatenRoh\3_Stockau_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\3_Stockau_X\*.dat Daten\3_Stockau
move DatenRoh\3_Stockau_X\*.Add Daten\3_Stockau
GOTO 2

:Sy
cd DatenRoh\3_Stockau_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\3_Stockau_Y\*.dat Daten\3_Stockau
move DatenRoh\3_Stockau_Y\*.Add Daten\3_Stockau
GOTO 2

rem mit*****
:M
cd DatenRoh\4_Mittelberg_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\4_Mittelberg_X\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_X\*.Add Daten\4_Mittelberg
cd DatenRoh\4_Mittelberg_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\4_Mittelberg_Y\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_Y\*.Add Daten\4_Mittelberg
GOTO 2

:Mx
cd DatenRoh\4_Mittelberg_X
DatAdd.exe %1 x %2 %3
cd ..\..
move v4_Mittelberg_X\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_X\*.Add Daten\4_Mittelberg
GOTO 2

:My
cd DatenRoh\4_Mittelberg_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\4_Mittelberg_Y\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_Y\*.Add Daten\4_Mittelberg
GOTO 2

rem pue*****
:P
cd DatenRoh\5_Puellersreuth_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_X\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_X\*.Add Daten\5_Puellersreuth
cd DatenRoh\5_Puellersreuth_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_Y\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_Y\*.Add Daten\5_Puellersreuth
GOTO 2

:Px
cd DatenRoh\5_Puellersreuth_X
DatAdd.exe %1 x %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_X\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_X\*.Add Daten\5_Puellersreuth
GOTO 2
cd DatenRoh\5_Puellersreuth_Y
DatAdd.exe %1 y %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_Y\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_Y\*.Add Daten\5_Puellersreuth
GOTO 2
```

Anhang II: Programmcode

```
rem alles*****
:ALL
cd DatenRoh\1_Berg_X
DatAdd.exe B x %2 %3
cd ..\..
move DatenRoh\1_Berg_X\*.dat Daten\1_Berg
move DatenRoh\1_Berg_X\*.Add Daten\1_Berg
cd DatenRoh\1_Berg_Y
DatAdd.exe B y %2 %3
cd ..\..
move DatenRoh\1_Berg_Y\*.dat Daten\1_Berg
move DatenRoh\1_Berg_Y\*.Add Daten\1_Berg

cd DatenRoh\2_Eiglasdorf_X
DatAdd.exe E x %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_X\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_X\*.Add Daten\2_Eiglasdorf
cd DatenRoh\2_Eiglasdorf_Y
DatAdd.exe E y %2 %3
cd ..\..
move DatenRoh\2_Eiglasdorf_Y\*.dat Daten\2_Eiglasdorf
move DatenRoh\2_Eiglasdorf_Y\*.Add Daten\2_Eiglasdorf

cd DatenRoh\3_Stockau_X
DatAdd.exe S x %2 %3
cd ..\..
move DatenRoh\3_Stockau_X\*.dat Daten\3_Stockau
move DatenRoh\3_Stockau_X\*.Add Daten\3_Stockau
cd DatenRoh\3_Stockau_Y
DatAdd.exe S y %2 %3
cd ..\..
move DatenRoh\3_Stockau_Y\*.dat Daten\3_Stockau
move DatenRoh\3_Stockau_Y\*.Add Daten\3_Stockau

cd DatenRoh\4_Mittelberg_X
DatAdd.exe M x %2 %3
cd ..\..
move DatenRoh\4_Mittelberg_X\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_X\*.Add Daten\4_Mittelberg
cd DatenRoh\4_Mittelberg_Y
DatAdd.exe M y %2 %3
cd ..\..
move DatenRoh\4_Mittelberg_Y\*.dat Daten\4_Mittelberg
move DatenRoh\4_Mittelberg_Y\*.Add Daten\4_Mittelberg

cd DatenRoh\5_Puellersreuth_X
DatAdd.exe P x %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_X\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_X\*.Add Daten\5_Puellersreuth
cd DatenRoh\5_Puellersreuth_Y
DatAdd.exe P y %2 %3
cd ..\..
move DatenRoh\5_Puellersreuth_Y\*.dat Daten\5_Puellersreuth
move DatenRoh\5_Puellersreuth_Y\*.Add Daten\5_Puellersreuth

GOTO 2

if not "%1" == "" GOTO 2

:1
echo.
echo DatAdd [Sta-
tion:'B','Bx','By','E','Ex','Ey','S','Sx','Sy','M','Mx','My','P','Px','Py','ALL']
[Start:jjddd] [End:jjddd]
echo.
:2
```

II.III Programmcode - *PreAnalyse.exe*

Der Einfachheit halber ist hier nicht der gesamte Programmcode dargestellt. Hier sind nur die Funktionen zu finden, welche unter 6.2.1.2 beschrieben sind. Zur besseren Übersicht sind zu Beginn nur die Namen und die übergebenen Parameter dargestellt.

```
namespace PreAnalyse
{
    public class WorkFunctions
    {
        public void AddOffset(string binpath, long i, float offset)
        public void LinearInterpolate(string binpath, long i, long j)
        public void Cut(string binpath, long i, long j)
        public void Fill(string binpath, long i)
        public void Auto_Interpolation(string binpath, string reportpath, string comment)
        public void UndoCutInterpolation(string binpath, float[] u, long i, long j)
    }
}
```

Im Folgenden ist die gesamte Klasse der modifizierenden Funktionen von *PraAnalse.exe* dargestellt.

```
using System;
using System.IO;
using System.Windows.Forms;

namespace PreAnalyse
{
    public class WorkFunctions
    {
        private LoadSave loadSave = null;

        public WorkFunctions()
        {
            loadSave = new LoadSave();
        }

        public void AddOffset(string binpath, long i, float offset)
        {
            loadSave.OpenBinary(binpath);
            // Manueller Offset mit vorher optisch festgelegtem Wert
            float u = 0;

            if(i != 0)
            {
                Progress progress = new Progress("Offset");
                progress.Show();
                while(i < MainWindow.indexLength) // Durchlaufen des Datensatzes
                {
                    u = loadSave.ReadSingle(i);
                    if(u < MainWindow.gap) //Test ist Wert ein Leerstelle dann keine Aktion
                    {
                        loadSave.SetPosition(i);
                        loadSave.binaryWriter.Write(u+offset);
                    }
                    i++; //Laufindex
                    progress.bar((int) (i*100/MainWindow.indexLength));
                }
                progress.Dispose();
            }
            else
                MessageBox.Show("Offset hier nicht möglich!", "PreAnalyse",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
            loadSave.CloseBinary();
        }
    }
}
```

```

}

public void LinearInterpolate(string binpath, long i, long j)
{
    loadSave.OpenBinary(binpath);
    //Lineare Interpolation zwischen i und j
    float sy=0;
    float sxy=0;
    short c=1;
    float sx=0;
    float beginValue=0;
    float endValue=0;
    float sx2=0;
    bool empty = false;
    // Interpolation mit Linearer Regression und Bereichsprüfung
    if( i > Parameter.Interpolation.Neighbourhods &&
        j < MainWindow.indexLength-Parameter.Interpolation.Neighbourhods &&
        Parameter.Interpolation.WithNeighbourhods )
    {
        //Bestimmung des ersten Wertes für Lineareinterpolation
        for (long a = i-Parameter.Interpolation.Neighbourhods ; a < i ; a++ )
        {
            loadSave.SetPosition(a);
            float y = loadSave.binaryReader.ReadSingle();
            if( y >= MainWindow.gap )//Test ob Regression Sinnvoll ist
                empty =true;
            sy += y;
            sxy +=c*y;
            sx += c;
            sx2 += c*c;
            c++;
        }
        if(!empty) //Extrapolation auf nächsten Wert bei positivem Test
        {
            c -=1;
            float B = (c*sxy - sx*sy) / (c*sx2 - sx*sx);
            float A = (sy - B*sx)/c;
            beginValue = A + B*(c+1);
        }
        else //Bestimmung des Wertes bei negativem Test
        {
            loadSave.SetPosition(i);
            beginValue = loadSave.binaryReader.ReadSingle();
            empty = false;
        }
        c=1;
        sy=0;
        sxy=0;
        sx=0;
        sx2=0;
        //Bestimmung des zweiten Wertes für Lineareinterpolation
        for(long b = j ; b <= j+Parameter.Interpolation.Neighbourhods ; b++)
        {
            loadSave.SetPosition(b);
            float y = loadSave.binaryReader.ReadSingle();
            if( y >= MainWindow.gap )//Test ob Regression Sinnvoll ist
                empty =true;
            sy += y;
            sxy +=c*y;
            sx += c;
            sx2 += c*c;
            c++;
        }
        if (!empty)//Extrapolation auf nächsten Wert bei positivem Test
        {
            c -=1;
            float B = (c*sxy - sx*sy) / (c*sx2 - sx*sx);
            float A = (sy - B*sx)/c;
            endValue = A+B ;
        }
        else //Bestimmung des zweiten Wertes bei negativem Test
        {
            loadSave.SetPosition(j);
            endValue = loadSave.binaryReader.ReadSingle();
        }
    }
}

```

```

    }
    else //Bestimmung der Werte für Interpolation wenn Nebenstellen nicht
          betrachtet werden sollen
    {
        loadSave.SetPosition(i);
        beginValue = loadSave.binaryReader.ReadSingle();
        loadSave.SetPosition(j);
        endValue   = loadSave.binaryReader.ReadSingle();
    }
    Progress progress = new Progress("Interpolation");

    // Test ob die Bestimmten Werte Leerstellen sind und Interpolation
    if( beginValue < MainWindow.gap && endValue < MainWindow.gap)
    {
        bool progressbar = false;
        if( j-i > 3000 )
        {
            progressbar = true;
            progress.Show();
        }
        Undo.CutInterpolation.index = i;
        if(MainWindow.files_both && !MainWindow.file1)
            Undo.CutInterpolation.u2 = new float[j-i+1];
        else
            Undo.CutInterpolation.u = new float[j-i+1];

        if(MainWindow.files_both && !MainWindow.file1)
        {
            for (int x = 0 ; x <= j-i ; x++)
            {
                Undo.CutInterpolation.u2[x] = loadSave.ReadSingle(i+x);
                if( progressbar )
                    progress.bar((int) (x*50/(j-i)));
            }
        }
        else
        {
            for (int x = 0 ; x <= j-i ; x++)
            {
                Undo.CutInterpolation.u[x] = loadSave.ReadSingle(i+x);
                if( progressbar )
                    progress.bar((int) (x*50/(j-i)));
            }
        }
        float m = (endValue - beginValue)/(j-i);
        loadSave.SetPosition(i+1);
        long k=i+1;
        float u;
        while(k < j )
        {
            u = beginValue + m*(k-i);
            loadSave.binaryWriter.Write(u);
            k++;
            if( progressbar )
                progress.bar((int) ((k+i)*50/j)+50);
        }
    }
    else // Ausgabe bei Interpolation auf Leerstellen
        MessageBox.Show("Interpolation nicht Möglich, da Anfaangs- oder Endpunkt
                        eine Leerstelle ist!", "PreAnalyse");
    loadSave.CloseBinary();
    progress.Dispose();
}

public void Cut(string binpath, long i, long j)
{
    loadSave.OpenBinary(binpath);
    //Werte zwischen i und j auf Leerstelle setzen
    Undo.CutInterpolation.index = i;
    if(MainWindow.files_both && !MainWindow.file1)
        Undo.CutInterpolation.u2 = new float[j-i+1];
    else
        Undo.CutInterpolation.u = new float[j-i+1];
    Progress progress = new Progress("Ausschneiden");
    bool progressbar = false;

```

```

        if( j-i > 3000 )
        {
            progressbar = true;
            progress.Show();
        }
        if(MainWindow.files_both && !MainWindow.file1)
            for (long x = 0 ; x <= j-i ; x++)
            {
                Undo.CutInterpolation.u2[x] = loadSave.ReadSingle(i+x);
                if( progressbar )
                    progress.bar((int)(x*50/(j-i)));
            }
        else
            for (long x = 0 ; x <= j-i ; x++)
            {
                Undo.CutInterpolation.u[x] = loadSave.ReadSingle(i+x);
                if( progressbar )
                    progress.bar((int)(x*50/(j-i)));
            }
        loadSave.SetPosition(i);
        long ibar = i;
        while(i++ <= j )
        {
            loadSave.binaryWriter.Write(MainWindow.gap);
            if( progressbar )
                progress.bar((int)((i+ ibar)*50/j)+50);
        }
        loadSave.CloseBinary();
        progress.Dispose();
    }

    public void Fill(string binpath,long i)
    {
        loadSave.OpenBinary(binpath);
        //Füllen von Lücken
        float a = 0;
        float b = 0;
        long j = i;

        a = loadSave.ReadSingle(i);
        b = a;
        //ersten Wert bestimmen
        while(a == MainWindow.gap)
        {
            i--;
            a = loadSave.ReadSingle(i);
            Undo.Fill.indexI = i;
            //Bereichsprüfung und Message
            if ( i == 0 && a == MainWindow.gap)
            {
                MessageBox.Show("Erster Wert ist eine Leerstelle.\n\nFüllen nicht  
möglich!", "PreAnalyse");
                return;
            }
        }
        //zweiten Wert bestimmen
        while(b == MainWindow.gap)
        {
            j++;
            if ( j > MainWindow.indexLength-1 )
                j = MainWindow.indexLength-1;
            b = loadSave.ReadSingle(j);
            Undo.Fill.IndexJ = j;
            //Bereichsprüfung und Message
            if ( j == MainWindow.indexLength-1 && b == MainWindow.gap)
            {
                MessageBox.Show("Letzter Wert ist eine Leerstelle.\n\nFüllen nicht  
möglich!", "PreAnalyse");
                return;
            }
        }
        loadSave.CloseBinary();
        //Lineare Interpolation mit den bestimmten Werten
        LinearInterpolate(binpath,i,j);
    }
}

```

```
public void Auto_Interpolation(string binpath, string reportpath, string comment)
{
    loadSave.OpenBinary(binpath);
    float    currentValue = 0f;
    float    passedValue  = 0f;
    //Reporteintrag und
    FileStream reportStream = null;
    StreamWriter reportWriter = null;
    reportStream = new FileStream(reportpath, FileMode.Append);
    reportWriter = new StreamWriter(reportStream);
    reportWriter.WriteLine("Automatische Interpolation" + comment);
    reportWriter.WriteLine("*****
    *****");

    Progress progress = new Progress("Autointerpolation");
    progress.Show();
    //Datensatz parsen
    for (long i=0; i < MainWindow.indexLength; i++)
    {
        passedValue = loadSave.ReadSingle(i); //Wert lesen
        if (i==0 && passedValue >= MainWindow.gap) //Falls erster Wert Leerstelle ist
            while (passedValue >= MainWindow.gap)
            {
                passedValue = loadSave.ReadSingle(i);
                i++;
            }
        progress.bar((int) (i*100/MainWindow.indexLength));
        if (passedValue >= MainWindow.gap) // Leerstelle suchen
        {
            long s=0;
            currentValue = passedValue;
            passedValue = loadSave.ReadSingle(i-1);
            //Endpunkt der Lücke suchen und Bereichstest
            while (currentValue >= MainWindow.gap && (i+s) <= (MainWindow.indexLength-1))
            {
                currentValue = loadSave.ReadSingle(i+s); //Wert Lesen
                s++; //Länge der Lücke
            }
            s -= 1;
            // Test ob Lücke klein genug ist
            if (s < Parameter.Interpolation.Count && (i+s) <= (MainWindow.indexLength-1))
            {
                //Lineare Regression mit Bereichstest
                if (i > Parameter.Interpolation.Neighbourhods &&
                    (i+s) < MainWindow.indexLength - Parameter.Interpolation.Neighbourhods && Parameter.Interpolation.WithNeighbourhods)
                {
                    //Anfangswert bestimmen mit angegebenen Nebenstellen
                    float sy=0;
                    float sxy=0;
                    short c=1;
                    float sx=0;
                    float sx2=0;
                    bool empty = false;
                    for (long a = i-Parameter.Interpolation.Neighbourhods; a < i; a++)
                    {
                        loadSave.SetPosition(a);
                        float y = loadSave.binaryReader.ReadSingle();
                        if (y >= MainWindow.gap) //Test ob Regression sinnvoll ist
                            empty = true;
                        sy += y;    sxy += c*y;    sx += c;    sx2 += c*c;    c++;
                    }
                    if (!empty) //Extrapolation des nächsten Wertes bei positivem Test
                    {
                        c -= 1;
                        float B = (c*sxy - sx*sy) / (c*sx2 - sx*sx);
                        float A = (sy - B*sx)/c;
                        passedValue = A + B*(c);
                        empty = false;
                    }
                    c=1; sy=0;    sxy=0; sx=0;    sx2=0;
                    // Endwert bestimmen mit angegebenen Nebenstellen
                    for (long b=i+s; b <= i+s+Parameter.Interpolation.Neighbourhods; b++)
                    {
```



```

        loadSave.SetPosition(b);
        float y = loadSave.binaryReader.ReadSingle();
        if (y >= MainWindow.gap) //Test ob Regression sinnvoll ist
            empty = true;
        sy += y;    sxy += c*y;    sx += c;    sx2 += c*c;    c++;
    }
    if (!empty) //Extrapolation des nächsten Wertes bei positivem Test
    {
        c -= 1;
        float B = (c*sxy - sx*sy) / (c*sx2 - sx*sx);
        float A = (sy - B*sx)/c;
        currentValue = A;
        empty = false;
    }
    //Interpolation mit den bestimmten Werten
    float dValue = (currentValue - passedValue) / s;
    passedValue += dValue;
    for (long inter = i; inter <= (i+s-1); inter++)
    {
        loadSave.SetPosition(inter);
        float r = passedValue + dValue*(inter-i);
        loadSave.binaryWriter.Write(r);
    }
    //Repteintrag
    reportWriter.WriteLine(" von " + loadSave.ReadTime(i));
    reportWriter.WriteLine(" bis "+loadSave.ReadTime(i+s) + " mit " + s +
        " Werten ersetzt.");
    reportWriter.WriteLine();
    }
    i += s; //Laufindex des Datensatzes
    progress.bar((int) (i*100/MainWindow.indexLength));
    }
}
progress.Dispose();
//Reporteintrag und Reportdatei schließen
reportWriter.WriteLine("*****");
reportWriter.Flush();
reportWriter.Close();
loadSave.CloseBinary();
}

public void UndoCutInterpolation(string binpath, float[] u, long i, long j)
{
    loadSave.OpenBinary(binpath);
    Progress progress = new Progress("Undo");
    progress.Show();
    int ibar = 0;
    long pos = i+1;
    loadSave.SetPosition(i);
    while(i++ <= j)
    {
        loadSave.binaryWriter.Write(u[i-pos]);
        progress.bar((int) ((i+ibar)*100/MainWindow.indexLength));
        ibar++;
    }
    progress.Dispose();
    loadSave.CloseBinary();
}
}
}

```

II.IV Programmcode - *EternaAdd.exe*

Um die Übersicht zu bewahren, ist von diesem Code nur die Hauptroutine dargestellt. Die verwendeten Funktionen sind nur mit Namen und den übergebenen Parametern zu finden, nicht aber entsprechende Quellcode.

```
using System;
using System.IO;
using System.Globalization;

namespace EternaAdd
{
    class Add
    {
        #region Globale Variablen
        private static string source1      = null;
        private static string source2      = null;
        private static string destinationfile = null;
        private static string reportfile    = null;
        private static FileStream sourcestream1      = null;
        private static StreamReader sourcereader1    = null;
        private static FileStream sourcestream2      = null;
        private static StreamReader sourcereader2    = null;
        private static FileStream destinationstream  = null;
        private static StreamWriter destinationwriter = null;
        private static FileStream reportstream      = null;
        private static StreamWriter reportwriter    = null;
        private static string str = null;
        private static string str1= null;
        private static DateTime currentTime = new DateTime();
        private static DateTime passedTime  = new DateTime();
        private static double currentValue = 0;
        private static double passedValue  = 0;
        private static double offset      = 0;
        private static long i_s1 = 0;
        private static long i_s2 = 0;
        private static long i_l  = 0;
        private static string Luecke = "99999.999";
        private static TimeSpan sample = new TimeSpan();
        public static NumberFormatInfo NumberFormat = new CultureInfo( "en-US",
                                false ).NumberFormat;

        #endregion
        [STAThread]
        static void Main(string[] args)
        {
            if (args.Length < 2)
            {
                showusage();
                return;
            }

            // Initialisierung der Parameter
            source1      = Environment.CurrentDirectory + "\\\" + args[0];
            source2      = Environment.CurrentDirectory + "\\\" + args[1];
            destinationfile = Environment.CurrentDirectory + "\\tmp.dat";
            reportfile    = Environment.CurrentDirectory + "\\\" +
                                Path.GetFileNameWithoutExtension(source1)+"_Rep.EAd";

            // Test auf vorhandensein der Quelldateien
            if(!FileExistCheck())
                return;
            // öffnen der Dateien
            open();
            // Kopieren des Headers der ersten Quelldatei
            copyHeader();

            // ermitteln der Samplerate und Test auf richtiges Datenformat
```

```

        if ((sample = readSamprlateSource1()) == new TimeSpan(0))
            return;

        // Formattest und kopieren der Daten der ersten Quelldatei
        if (!copyValue(sourcereader1, source1, offset))
            return;
        // Parameter setzen für Anzahl der enthaltenen Werte
        i_s1 += i_s2; i_s2 = 0;

        // überlesen des Headers der zweiten Quelldatei
        for (int i = 0 ; i < 14 ; i++)
            str = sourcereader2.ReadLine();

        // ermitteln des ersten Wertes und Zeitkennung + Farmattest
        if (!readFirstValueSource2())
            return;

        // Verhalten der verschiedenen Möglichkeiten für Übergänge zwischen Datenreihen
        between();

        // kopieren der Werte der zweiten Quelldatei
        if (!copyValue(sourcereader2, source2, offset))
            return;

        // Reportausgabe
        Console.WriteLine("\nAn Datei: " + Path.GetFileName(source1)+ " mit " + i_s1 +
            " Werten wurde\n    Datei: " + Path.GetFileName(source2)+ " mit "
            " + i_s2 + " Werten angehaengt!");
        Console.WriteLine("Datei: " + Path.GetFileName(source1) + " enthaelt " +
            (i_s1+i_s2) + " Werte ");
        reportwriter.WriteLine("\nSamplerate: " + sample.Hours.ToString("#00") + ":" +
            sample.Minutes.ToString("#00") + ":" + sample.Seconds.ToString("#00") );
        reportwriter.WriteLine("\nAn Datei: " + Path.GetFileName(source1)+ " mit " +
            i_s1 + " Werten wurde\n    Datei: " + Path.GetFileName(source2)+ " mit " +
            i_s2 + " Werten angehaengt!");
        reportwriter.Write("Datei: " + Path.GetFileName(source1) + " enthaelt " +
            (i_s1+i_s2) + " Werte ");
        if (i_l < 0)
        {
            Console.WriteLine("\n    und " + Math.Abs(i_l) + " wurden am Anfang der
                Datei: " + Path.GetFileName(source2) + " entfernt!\n");
            reportwriter.WriteLine("\n    und " + Math.Abs(i_l) + " wurden am Anfang der
                Datei: " + Path.GetFileName(source2) + " entfernt!\n");
        }
        if (i_l > 0)
        {
            Console.WriteLine("\n    und "+i_l+" wurden zwischen den Dateien
                hinzugefügt!\n");
            reportwriter.WriteLine("\n    und " + i_l +
                " wurden zwischen den Dateien hinzugefügt!\n");
        }
        if (i_l == 0)
            Console.WriteLine("\n");
        // schließen der Dateien
        close();
    }

    #region Funktionen
    private static void showusage()
    private static bool FileExistCheck()
    private static void open()
    private static void close()
    private static void failureClose()
    private static void copyHeader()
    private static DateTime ConvertToDateTime(string str)
    private static TimeSpan readSamprlateSource1()
    private static bool readFirstValueSource2()
    private static bool copyValue(StreamReader source, string sourcefile,
        double offset)
    private static bool timeCheck()
    private static void between()
    private static string format(double v)
    #endregion
}
}

```

II.V Programmcode - *Filter.exe*

Der Programmcode von *Filter.exe* ist folgend abgedruckt.

```
using System;
using System.IO;
using System.Globalization;

namespace Filter
{
    class Filter
    {
        #region File Variablen
        private static string sourcefile      = null;
        private static string destinationfile = null;
        private static string filterfile      = null;
        private static string residuumfile    = null;

        private static FileStream sourcestream      = null;
        private static StreamReader sourcereader    = null;
        private static FileStream destinationstream = null;
        private static StreamWriter destinationwriter = null;
        private static FileStream filterstream      = null;
        private static StreamReader filterreader    = null;
        private static FileStream residuumstream    = null;
        private static StreamWriter residuumwriter  = null;
        #endregion

        #region Filtervariablen
        public static double[] FilterField = null;
        public static int FilterLength = 0;
        public static int StuetzstellenAbstand = 0;
        public static int AnzahlFilterElemente = 0;
        public static double Faktor = 0;
        public static int StutzstellenInput = 0;
        public static int StutzstellenOutput = 0;
        public static float Prozent = 0;
        public static double Lueckenkennung = 0;
        #endregion

        #region Globale Variablen
        public static NumberFormatInfo NumberFormat = new CultureInfo( "en-US",
                                                                    false ).NumberFormat;

        public static double[] FilterValue = null;
        private static double[] arrayTmp = null;
        private static double[] arrayLastHour = null;

        public static double filteredValue = 0;
        public static DateTime filterPoint = new DateTime();
        private static DateTime currentTime = new DateTime();
        private static DateTime passedTime  = new DateTime();
        private static TimeSpan SampleRate = new TimeSpan();
        private static TimeSpan shiftf = new TimeSpan();

        private static double unFilteredValue = 0;

        public static long Length = 0;
        private static long gefilterteWerte = 0;
        private static long gefilterteWerteGesamt = 0;
        private static long Werte = 1;
        private static int lueckeEndecount = 0;
        private static int lueckeAnfangcount = 0;

        private static bool EOF = false;

        private static bool LueckeAnfang = false;
        private static bool LueckeAnfangWrite = false;
        private static bool lueckeEnde = false;
        private static bool LueckeEndeWrite = false;
        private static bool Luecke = false;
        private static bool Lueckel = false;
        }
```

```

private static int c = 0;
private static double residuum = 0;
private static int shift = 0;
#endregion

[STAThread]
static void Main(string[] args)
{
    if (args.Length < 2)
    {
        showusage();
        return;
    }
    sourcefile = Environment.CurrentDirectory + "\\\" + args[0];
    destinationfile = Environment.CurrentDirectory + "\\\" + args[1];
    filterfile = Environment.CurrentDirectory + "\\Filter.dat";
    residuumfile = Environment.CurrentDirectory + "\\\" +
        Path.GetFileNameWithoutExtension(destinationfile) + ".ReJ";

    string space = null;
    for (int i = 0 ; i < 38 - Path.GetFileName(sourcefile).Length;i++)
        space += " ";
    Console.WriteLine();
    Console.WriteLine("*****");
    Console.WriteLine("* Filtern der Datei: "
        +Path.GetFileName(sourcefile)+space+ "(c) A.G. Ver.: 1.0 *");
    Console.WriteLine("*****");
    Console.WriteLine();

    //Test ob entstreichende Dateiecn Existieren
    if (FileExistCheck())
        return;

    //Filterparameter lesen
    ReadFilter();
    //Quell-, Ziel- und Residuumdatei
    open();
    //Header für Ziel- und Residuumdatei Kopieren
    CopyHeader();

    //Samplerate des Inputfiles
    SampleRate = new TimeSpan(((long) (StutzstellenInput)) *10000000);

    //Zeitverschiebung zum nächsten Filterpunkt (Samplerate des Outputfiles)
    shiftf = new TimeSpan(((long) (StutzstellenOutput)) *10000000);

    //Abbruchkriterium aus Filterlänge/2 für Lückenbetrachtung
    c = FilterLength/2/(StutzstellenOutput/StutzstellenInput);

    //Indexshift Berechnung
    shift = StutzstellenOutput / StutzstellenInput;

    //Doublearray für Filterung anlegen
    FilterValue = new double[FilterLength];

    //Doublearray für Filterung an Lücken oder am Ende der Zeitreihe anlegen
    arrayTmp = new double[FilterLength];

    //Doublearray für Filterung an Lücke anlegen
    arrayLastHour = new double[FilterLength/2/c];

    //temporärer String zum Zeilenweisen lesen der Quelldatei
    string str = null;

    //erste Zeile im Datensatz lesen um Timecheck ausüben zu können und Test ob
    erster Wert
        Leerstelle ist oder Anfangszeit nicht stimmt
    bool erster = false;
    while(!erster)
    {
        //Initialisierung des Indexes zum lesen des ersten Wertes
        Length = FilterLength / 2;

```

```

        str = sourcereader.ReadLine();
        if( str != null && str.Length >= 1)
            getFilterValues(str);
        //Leerstellentest und Anfangszeittest
        if ((FilterValue[FilterLength/2] != Lueckenkennung) &&
            ((currentTime.Hour*3600 + currentTime.Minute*60 +
              currentTime.Second) % ((long)(StutzstellenOutput)) == 0))
            erster = true;
    }
    filterPoint = currentTime;
    passedTime = currentTime;

    Console.WriteLine("Filter fortschritt:");

    //Scanroutine und Timecheck
    while ( !EoF ) //Ausführen bis Quelldateiende
    {
        //Filterarray füllen
        while(Length != FilterLength && EoF == false)
        {
            //Zeile aus Quelldatei Lesen
            str = sourcereader.ReadLine();
            if( str != null && str.Length >= 1)
            {
                // Filterwert und Gehalt ermitteln
                getFilterValues(str);
                // Zeitcheck ob die Zeitreihe auch in sich schlüssig ist
                if (Timecheck(passedTime,currentTime,SampleRate))
                    return;
                //Zeitumschreiben für nächsten Durchlauf
                passedTime = currentTime;
                Werte++;
            }
            else
                EoF = true; // Marke für Dateiende setzten und behandlung auslösen
        }

        gefilterteWerte++;
        gefilterteWerteGesamt++;

        // Test und Behandlung von Lücken
        LueckeEndeBerechnung();
        LueckeAnfangBerechnung();

        //Berechnung des Filterwertes
        filteredValue = CalculateFilterValue();

        residuum = FilterValue[FilterLength / 2] - filteredValue;
        RahmenbedingungFürLuecke();

        //Output
        WriteOutput(filterPoint,filteredValue,residuum);

        //Zeitreihe um einen Filterwert verschieben
        for (int i = 0; i < FilterLength - shift; i++)
            FilterValue[i] = FilterValue[i+shift];
        //Berechnung neuer Filterzeitpunkt
        filterPoint = filterPoint + shiftf;
        //Indexvariable verschieben
        Length = Length - shift;

        if(gefilterteWerteGesamt % 84 == 0)
            Console.WriteLine("");
    }
    if ( EoF )
        Eof();
    close();
    space = " ";
    string spaceQ = " ";
    string spaceZ = " ";

    if( Path.GetFileName(sourcefile).Length >
        Path.GetFileName(destinationfile).Length)

```

```

    {
        for(int i = 0; i < Path.GetFileName(sourcefile).Length; i++)
            space += " ";
        for(int i = 0; i < Path.GetFileName(sourcefile).Length -
            Path.GetFileName(destinationfile).Length; i++)
            spaceZ += " ";
    }
    else
    {
        for(int i = 0; i < Path.GetFileName(destinationfile).Length; i++)
            space += " ";
        for(int i = 0; i < Path.GetFileName(destinationfile).Length -
            Path.GetFileName(sourcefile).Length; i++)
            spaceQ += " ";
    }
    Console.WriteLine("\n\n" + space + "Samplerate Werte" );
    Console.WriteLine( "Quelldatei : " + Path.GetFileName(sourcefile) + spaceQ +
        SampleRate.Hours.ToString("#00") + ":" + SampleRate.Minutes.ToString("#00") + ":" + SampleRate.Seconds.ToString("#00") +
        " + Werte + "\n" + "Zieldatei : " + Path.GetFileName(destinationfile) +
        spaceZ + shiftf.Hours.ToString("#00") + ":" + shiftf.Minutes.ToString("#00") + ":" + shiftf.Seconds.ToString("#00") +
        " + gefilterteWerteGesamt + "\n" + "Reject-Datei: " +
        Path.GetFileName(residuumfile) + spaceZ + shiftf.Hours.ToString("#00") +
        ":" + shiftf.Minutes.ToString("#00") + ":" + shiftf.Seconds.ToString("#00") +
        " + gefilterteWerteGesamt + "\n");
}

#region Funktionen
private static void getFilterValues(string str)
{
    currentTime = ConvertToTime(str);
    unFilteredValue = Convert.ToDouble(str.Substring(15, 10), NumberFormat);

    if ( unFilteredValue == Lueckenkennung)
    {
        Luecke = true;
        LueckeAnfang = true;
    }
    if ( unFilteredValue != Lueckenkennung && LueckeAnfang && !Luecke)
    {
        LueckeAnfang = false;
        lueckeEnde = true;
    }

    FilterValue[Length] = unFilteredValue * Faktor;

    Length++;
}
private static void LueckeEndeBerechnung()
{
    // Ende Datenreihe vor Lücke
    if ( !LueckeAnfang && !LueckeAnfangWrite)
        for ( int i = 0; i < FilterLength/2/c; i++)
            arrayLastHour[i] = FilterValue[FilterLength-2-i];

    if ( LueckeAnfang )
        LueckeAnfangWrite = true;

    if (LueckeAnfangWrite)
    {
        if (lueckeAnfangcount == 0)
        {
            //Zeitreihe kopieren bis Lücke
            gefilterteWerte = gefilterteWerte;
            for ( int i = 0 ; i < FilterLength - (FilterLength/2/c); i++)
                arrayTmp[i] = FilterValue[i];
            //Letzte gute Stunde kopieren und spiegeln
            for ( int i = 0 ; i < (FilterLength/2/c); i++)
                arrayTmp[FilterLength-(FilterLength/2/c) +i] = arrayLastHour[i];
        }
        if(lueckeAnfangcount != 0 && lueckeAnfangcount <= c) //-1
        {
            //Zeitreihe kopieren bis Lücke

```

```

        for ( int i = 0 ; i < FilterLength -
              (FilterLength/2/c*(lueckeAnfangcount+1)); i++)
            arrayTmp[i] = FilterValue[i];

        //Stunden Spiegeln bis letzte Stunde
        for ( int i = 0 ; i < (FilterLength/2/c*(lueckeAnfangcount+1)) ; i++)
            arrayTmp[FilterLength - (FilterLength/2/c*(lueckeAnfangcount+1)) + i]
                = FilterValue[FilterLength-2-
                              (FilterLength/2/c*(lueckeAnfangcount+1))-i];
    }
    lueckeAnfangcount++;
}

if (LueckeAnfangWrite && lueckeAnfangcount == c )
{
    LueckeAnfangWrite = false;
    lueckeAnfangcount = 0;
    Luecke1 = true;
}

private static void LueckeAnfangBerechnung()
{
    if (lueckeEnde) // markierung setzen für Lückenende
    {
        LueckeEndeWrite = true;
        lueckeEnde = false;
    }
    if( LueckeEndeWrite)
        lueckeEndecount++;

    if (LueckeEndeWrite && lueckeEndecount == c)
    {
        lueckeEndecount = 0;
        gefilterteWerte = 1;
        LueckeEndeWrite = false;
        Luecke1 = false;
    }
    if (gefilterteWerte == 1) //Spiegeln
    {
        for (int i=0;i<FilterLength / 2; i++)
            FilterValue[FilterLength/2 - i]=FilterValue[FilterLength/2 + i];
    }
}

private static void RahmenbedingungFürLuecke()
{
    int x = (int) (Math.Ceiling(((float) (FilterLength/2)) * Prozent/100 /
                              ((float) (StutzstellenOutput/StutzstellenInput)) ));

    //Rand Anfang
    if(c+1-lueckeAnfangcount <= x )
    {
        filteredValue = Lueckenkennung;
        residuum = Lueckenkennung;
    }

    //Mitte
    if (Luecke1)
    {
        filteredValue = Lueckenkennung;
        residuum = Lueckenkennung;
    }

    //Rand Ende
    if (gefilterteWerte <= x)
    {
        filteredValue = Lueckenkennung;
        residuum = Lueckenkennung;
    }

    //if(Luecke )
    Luecke = false;
}

```



```

private static void open()
{
    //Input File Öffnen
    sourcestream = new FileStream(sourcefile, FileMode.Open);
    sourcereader = new StreamReader(sourcestream);
    //Output File Öffnen
    destinationstream = new FileStream(destinationfile, FileMode.Create);
    destinationwriter = new StreamWriter(destinationstream);
    //Residuum File Öffnen
    residuumstream = new FileStream(residuumfile, FileMode.Create);
    residuumwriter = new StreamWriter(residuumstream);
}

private static void close()
{
    //Input File schließen
    sourcereader.Close();
    //Output File zu Ende schreiben und schließen
    destinationwriter.Flush();
    destinationwriter.Close();
    //Residuum File zu Ende schreiben und schließen
    residuumwriter.Flush();
    residuumwriter.Close();
}

private static void CopyHeader()
{
    //Header lesen
    string[] Header = new string[14];
    for (int i=0; i<14; i++)
        Header[i] = sourcereader.ReadLine();

    //Header für Gefiltertem Output schreiben
    for (int i = 0; i<9; i++)
        destinationwriter.WriteLine(Header[i]);
    destinationwriter.WriteLine(Header[9] + " Gefiltert");
    for (int i = 10; i<14;i++)
        destinationwriter.WriteLine(Header[i]);

    //Header für Residuum schreiben
    for (int i = 0; i<9; i++)
        residuumwriter.WriteLine(Header[i]);
    residuumwriter.WriteLine(Header[9] + " Residuum");
    for (int i = 10; i<14;i++)
        residuumwriter.WriteLine(Header[i]);
}

private static void ReadFilter()
{
    filterstream = new FileStream(filterfile, FileMode.Open);
    filterreader = new StreamReader(filterstream);

    string str = filterreader.ReadLine();
    str = filterreader.ReadLine();
    FilterLength = Convert.ToInt32(str.Substring(27, str.Length-27));
    str = filterreader.ReadLine();
    StuetzstellenAbstand = Convert.ToInt32(str.Substring(27, str.Length-27));
    str = filterreader.ReadLine();
    AnzahlFilterElemente = Convert.ToInt32(str.Substring(27, str.Length-27));

    for (int i=0;i<3;i++)
        filterreader.ReadLine();

    str = filterreader.ReadLine();
    Faktor = Convert.ToDouble(str.Substring(72, str.Length-72), NumberFormat);
    str = filterreader.ReadLine();
    StutzstellenInput = Convert.ToInt32(str.Substring(72, str.Length-
        72), NumberFormat);

    str = filterreader.ReadLine();
    StutzstellenOutput = Convert.ToInt32(str.Substring(72, str.Length-
        72), NumberFormat);

    str = filterreader.ReadLine();
    Prozent = Convert.ToSingle(str.Substring(72, str.Length-72), NumberFormat);
    str = filterreader.ReadLine();
    Lueckenkennung= Convert.ToDouble(str.Substring(72, str.Length-72), NumberFormat);
}

```

```

        for (int i=0;i<8;i++)
            filterreader.ReadLine();

        Console.WriteLine("Lesen der Filterparameter");
        Console.WriteLine();

        FilterField = new double[FilterLength];

        for(int i= 0;i < FilterLength;i++)
        {
            str = filterreader.ReadLine();
            if (Convert.ToInt32(str.Substring(0,6))-1 != i)
            {
                Console.WriteLine("Der Filteroperator ist nicht Kosistent in Zeile: "+i);
                return;
            }
            string st = str.Substring(6,str.Length-6);
            FilterField[i] = Convert.ToDouble(st ,NumberFormat);
        }
        filterreader.Close();
    }

    private static bool Timecheck(DateTime passedTime, DateTime currentTime,
        TimeSpan SampleRate)
    {
        TimeSpan x = (currentTime - passedTime);
        if ( x != SampleRate )
        {
            Console.WriteLine("Zeitmarke bei Zeitpunkt " + currentTime +
                " nicht in Ordnung oder stimmt nicht mit den Fileterparametern uebe-
                rein!");
            return true;
        }
        else
            return false;
    }

    private static double CalculateFilterValue()
    {
        double fValue = 0;
        if ( !LueckeAnfangWrite)
        {
            for (int i=0; i < FilterLength; i++)
                fValue = fValue + (FilterValue[i] * FilterField[i]);
        }
        else
        {
            //normale Berechnung
            for (int i=0; i < FilterLength; i++)
                fValue = fValue + (arrayTmp[i] * FilterField[i]);
        }
        return fValue;
    }

    private static void Eof()
    {
        lueckeAnfangcount = 1;
        LueckeAnfangWrite = true;
        int x = (int)(Math.Ceiling(((float)(FilterLength/2)) * Prozent/100 /
            ((float)(StutzstellenOutput/StutzstellenInput)) ));
        //abzug desüberschüssigen Zeitpunktes
        filterPoint = filterPoint - shiftf;

        for (int i = 0; i < c-x-2 ;i++)
        {
            for ( int j = 0 ; j < FilterLength; j++)
                arrayTmp[j] = 0;
            //Zeitreihe kopieren bis Lücke
            for ( int j = 0 ; j < FilterLength - (FilterLength/2/c*(lueckeAnfangcount));
j++)
                arrayTmp[j] = FilterValue[j+shift*(i+1)];

            //Stunden Spiegeln bis letzte Stunde
            for ( int j = 0 ; j < (FilterLength/2/c*(lueckeAnfangcount)) ; j++)
                arrayTmp[FilterLength - (FilterLength/2/c*(lueckeAnfangcount)) + j]

```

```

        = FilterValue[FilterLength - 2 - (Filter-
        Length/2/c*(lueckeAnfangcount)) - j];
//Berechnung des Filterwertes
filteredValue = CalculateFilterValue();
//Berechnung des Zeitpunktes des entsprechenden Filterwertes
filterPoint = filterPoint + shiftf;

    residuum = arrayTmp[FilterLength/2] - filteredValue;
    WriteOutput(filterPoint,filteredValue,residuum);
    lueckeAnfangcount++;
    gefilterteWerteGesamt++;
}
}

#endregion
#region Hilfsroutinen
private static void showusage()
{
    Console.WriteLine();
    Console.WriteLine("*****");
    Console.WriteLine("  Das Programm benötigt 2 Eingabe Argumente!      *");
    Console.WriteLine("  * Filter [Quelldatei] [Zielfilter]                *");
    Console.WriteLine("  * Filter.exe ein Programm das Dateien im Eternaformat *");
    Console.WriteLine("  * filtert.                                           *");
    Console.WriteLine("  * Dazu ist eine Filterparameterdatei 'Filter.dat' notwendig. *");
    Console.WriteLine("  * Diese*");
    Console.WriteLine("  * muss sich im Arbeitsverzeichnis befinden!      *");
    Console.WriteLine("  * André Gebauer                                     *");
    Console.WriteLine("*****");
    Console.WriteLine();
}

private static bool FileExistCheck()
{
    if(!File.Exists(sourcefile))
    {
        Console.WriteLine("Quelldatei existiert nicht!");
        return true;
    }
    if(!File.Exists(filterfile))
    {
        Console.WriteLine("Filter existiert nicht!");
        return true;
    }

    //else
    return false;
}

private static string format(double v)
{
    string tmpvalue = v.ToString("0.000", NumberFormat);
    string tmpspace = null;
    for (short x=0;x < 10-tmpvalue.Length;x++)
        tmpspace += " ";
    return tmpspace+tmpvalue;
}

private static DateTime ConvertToTime(string str)
{
    int year, month, day, hour, minute, second;
    try
    {
        year = Convert.ToInt32(str.Substring(0, 4));
    }
    catch(Exception)
    {
        year = 0;
    }
    try
    {
        month = Convert.ToInt32(str.Substring(4, 2));
    }

```

```

    }
    catch (Exception)
    {
        month = 0;
    }
    try
    {
        day = Convert.ToInt32(str.Substring(6, 2));
    }
    catch
    {
        day = 0;
    }
    try
    {
        hour = Convert.ToInt32(str.Substring(9, 2));
    }
    catch (Exception)
    {
        hour = 0;
    }
    try
    {
        minute = Convert.ToInt32(str.Substring(11, 2));
    }
    catch (Exception)
    {
        minute = 0;
    }
    try
    {
        second = Convert.ToInt32(str.Substring(13, 2));
    }
    catch (Exception)
    {
        second = 0;
    }

    return(new DateTime(year, month, day, hour, minute, second));
}
private static void WriteOutput(DateTime filterPoint, double filteredValue,
                                double residuum)
{
    destinationwriter.WriteLine(filterPoint.Year.ToString("#0000") +
        filterPoint.Month.ToString("#00") + filterPoint.Day.ToString("#00") +
        " " + filterPoint.Hour.ToString("#00") + filter-
        Point.Minute.ToString("#00") + filterPoint.Second.ToString("#00") +
        format(filteredValue));
    residuumwriter.WriteLine(filterPoint.Year.ToString("#0000") +
        filterPoint.Month.ToString("#00") + filterPoint.Day.ToString("#00") +
        " " + filterPoint.Hour.ToString("#00") + filter-
        Point.Minute.ToString("#00") + filterPoint.Second.ToString("#00") +
        format(residuum));
}
#endregion
}
}

```

II.VI Programmcode - *Kalibrierung.exe*

Da die grobe Struktur der verwendeten Programme bei allen Konsolenanwendungen in etwa gleich sind, ist nur die Routine dargestellt, welche die Kalibrierung vornimmt. Die Bedeutung der verwendeten Variablen lässt sich aus den zuvor dargestellten Programmen erschließen.

```
while( (str = sourcereader.ReadLine()) != null && str.Length >= 1)
{
    if(str.Substring(15, 10) == gap)
    {
        destinationwriter.WriteLine(str.Substring(0,15)+leerstelle);
        count++;
    }
    else
    {
        v = Convert.ToSingle(str.Substring(15, 10), NumberFormat)*cal;
        destinationwriter.WriteLine(str.Substring(0,15)+format(v));
        count++;
    }
}
```

II.VII Programmcode - *Transform.exe*

Wie in Anhang II.V ist hier wiederum nur ein Auszug aus der Hauptroutine dargestellt.

```
//Transformation für alle Werte der Datensätze
strX = sourcereaderX.ReadLine();
strY = sourcereaderY.ReadLine();
while( (strX != null) && (strX.Length >= 1) || (strY != null) && (strY.Length >= 1) )
{
    //Test für gleiche Länge der Datensätze
    if (strX != null && strY != null )
    {
        if(strX.Length == 25 && strY.Length == 25)
        {
            //Test ob in beiden Datensätzen Lücken sind
            if((strX.Substring(15, 10) == gap)|| (strY.Substring(15, 10) == gap))
            {
                destinationwriterNS.WriteLine(strX.Substring(0,15)+gap);
                destinationwriterEW.WriteLine(strY.Substring(0,15)+gap);
                count++;
            }
            else
            {
                //Transformation und Speichern
                X = Convert.ToSingle(strX.Substring(15, 10), NumberFormat);
                Y = Convert.ToSingle(strY.Substring(15, 10), NumberFormat);

                SN = X*(float) (Math.Sin(azimut))+Y*(float) (Math.Cos(azimut));
                EW = (X*(float) (Math.Cos(azimut))-Y*(float) (Math.Sin(azimut)))*(-1);
                destinationwriterNS.WriteLine(strX.Substring(0,15) + format(SN));
                destinationwriterEW.WriteLine(strY.Substring(0,15) + format(EW));
                count++;
            }
        }
        else
        {
            //Dateien fertig schreiben und schließen
            close();
            //Meldung nach Beendigung der Transformation
        }
    }
}
```

```

        Console.WriteLine("\nDateien \n " + sourceX + " und " + sourceY + " auf \n "
            + destNS + " und " + destEW + " \nmit "
            + count + " Werten Transformatiert!\nAzimut: " + azimutstring);
        //Meldung bei Formatfehlern
        Console.WriteLine("\n*** Prüfen ob Format der Dateien richtig ist! ***\n");
        return;
    }
}
else
{
    //Dateien fertig schreiben und schließen
    close();
    //Meldung nach Beendigung der Transformation
    Console.WriteLine("\nDateien \n " + sourceX + " und " + sourceY + " auf \n "
        + destNS + " und " + destEW + " \nmit "
        + count + " Werten Transformatiert!\nAzimut: " + azimutstring);
    //Meldung bei unterschiedlich langen Datensätzen
    Console.WriteLine(
        "\n*** Datensätze waren unterschiedlich lang und wurden angeglichen! ***\n");
    return;
}
strX = sourcereaderX.ReadLine();
strY = sourcereaderY.ReadLine();
}

```

III Steuerfile für *Analyse.exe*

Nachfolgend ist ein Steuerfile für die Gezeitenanalyse mit dem Programm ANALYZE aus dem ETERNA 3.30 Paket aufgelistet. Mit diesem File erfolgt die Analyse von Neigungszeiten der Ostwest-Komponente der Station Berg. Der Sensorname ist frei wählbar und hier mit „Pendel“ festgelegt, danach folgen der Stützstellenabstand in den Rohdaten von einer Stunde und die geographischen Daten der Station. Unter Stationsazimut gibt man die Himmelsrichtung an, in der registriert wurde und legt anschließend Beobachtungsbeginn, zu analysierende Komponente (1=Neigung), Gezeitenkatalog und Genauigkeit fest. Dann folgt ein Block mit den theoretischen Parametern der einzelnen zu untersuchenden Wellen, ein Block mit den verwendeten Filtern und einer mit Ausgabeparametern, bevor man eventuelle Korrelationen mit meteorologischen Parametern festlegen kann.

```

# This file ASKN1997.INI status 2000.06.25 containing control parameters
# for programs PREGRED 3.30 and ANALYZE 3.30

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# ! NOTE: The datalines have to start with their names.                !
# !       An additional comment may follow after the values,          !
# !       delimited by a whitespace                                    !
# ! Data input file must be in order: Date, Time, Channel n          !
# !       with the following format: I8,I1X,I6,4F10.3                  !
# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# a commentline starts with an '#', it may appear at any position
# in this file. Empty lines may appear too

TEXTHEADER= Pendel ASK103 Berg
TEXTHEADER= Hourly Values

```

Anhang III: Steuerfile für Analyse.exe

```

TEXTHEADER= Corrected
TEXTHEADER= calibrated
TEXTHEADER= Kanal - EW
TEXTHEADER=
TEXTHEADER=
TEXTHEADER=
TEXTHEADER=
TEXTHEADER=

SENSORNAME= Pendel #earth tide sensor name
SAMPLERATE= 3600 #sampling interval in seconds
STATLATITU= 49.813404 #stations latitude in degree(S=neg.)
STATLONITU= 12.141137 #stations longitude in degree(E=pos.)
STATELEVAT= 300.000 #stations elevation in meter
STATGRAVIT= 0. #stations gravity in m/s**2
STATAZIMUT= 270. #stations azimuth in degree from north
INITIALEPO= 2003 11 04 #initial epoch in year,month,day,hour
TIDALCOMPO= 1 #tidal component, see manual
TIDALPOTEN= 7 #tidal potential catalogue (2=CTE)
AMTRUNCATE= 1.D-7 #amplitude threshold for tidal potential
STEPDETLIM= 5. #DESPIKE limit for step detection (nm/s**2)
SPIKDETLIM= 2. #DESPIKE limit for spike detection (nm/s**2)

TIDALPARAM= 0.501370 0.911390 1.14693 -0.1689 Q1 #tidal param.
TIDALPARAM= 0.911391 0.947991 1.14768 0.0614 O1 #tidal param.
TIDALPARAM= 0.947992 0.981854 1.14009 0.1437 M1 #tidal param.
TIDALPARAM= 0.981855 1.023622 1.13592 0.2426 K1 #tidal param.
TIDALPARAM= 1.023623 1.057485 1.15568 0.2584 J1 #tidal param.
TIDALPARAM= 1.057486 1.470243 1.15747 0.1085 OO1 #tidal param.
TIDALPARAM= 1.470244 1.880264 1.15102 2.5392 2N2 #tidal param.
TIDALPARAM= 1.880265 1.914128 1.16962 2.4398 N2 #tidal param.
TIDALPARAM= 1.914129 1.950419 1.18554 1.9894 M2 #tidal param.
TIDALPARAM= 1.950420 1.984282 1.21379 3.3195 L2 #tidal param.
TIDALPARAM= 1.984283 2.451943 1.18722 0.5274 S2 #tidal param.
TIDALPARAM= 2.451944 7.000000 0.16594 146.9661 M3M6 #tidal param.

PRINTDEVEL= 1 #ANALYZE print param. for tidal development (1=yes)
SEARDA TLIM= -1. #ANALYZE search for data error threshold
NUMHIGPASS= 1 #ANALYZE highpassfiltering = 1
NUMFILNAME= nlhlh002.nlf #ANALYZE numerical filter no. 7
PRINTOBSER= 0 #ANALYZE print parameter for observations (1=yes)
PRINTLFOBS= 0 #ANALYZE print parameter fo lowpass filtered obs.
RIGIDEARTH= 0 #ANALYZE parameter for rigid earth model (1=yes)
HANNWINDOW= 1 #ANALYZE parameter for Hann-window (1=yes)
QUICKLOOKA= 0 #ANALYZE parameter for quick look analysis (1=yes)

WAVEGROUPI= 0.501370 0.911390 1.00000 0.0000 Q1 #ETERNA wavegroup
WAVEGROUPI= 0.911391 0.947991 1.00000 0.0000 O1 #ETERNA wavegroup
WAVEGROUPI= 0.947992 0.981854 1.00000 0.0000 M1 #ETERNA wavegroup
WAVEGROUPI= 0.981855 1.023622 1.00000 0.0000 K1 #ETERNA wavegroup
WAVEGROUPI= 1.023623 1.057485 1.00000 0.0000 J1 #ETERNA wavegroup
WAVEGROUPI= 1.057486 1.470243 1.00000 0.0000 OO1 #ETERNA wavegroup
WAVEGROUPI= 1.470244 1.880264 1.00000 0.0000 2N2 #ETERNA wavegroup
WAVEGROUPI= 1.880265 1.914128 1.00000 0.0000 N2 #ETERNA wavegroup
WAVEGROUPI= 1.914129 1.950419 1.00000 0.0000 M2 #ETERNA wavegroup
WAVEGROUPI= 1.950420 1.984282 1.00000 0.0000 L2 #ETERNA wavegroup
WAVEGROUPI= 1.984283 2.451943 1.00000 0.0000 S2 #ETERNA wavegroup
WAVEGROUPI= 2.451944 7.000000 1.00000 0.0000 M3M6 #ETERNA wavegroup

#METEOPARAM= 1 3.56hPa #METEOROLOGIE

HPERPAGE=24 #PREGRED displayed hours per page

PGUPPGDN=12 #New hours per Pageup / Pagedown

RANGETC1=8000 #PREGRED range for Channel 1 in nm/s^2
RANGETC2=30 #PREGRED range for Channel 2 in hPascal
RANGETC3=2500 #PREGRED range for Channel 3 in nm/s^2
RANGETC4=250 #PREGRED range for Channel 4 in nm/s^2

RANGRAW1=2 #PREGRED range for Channel 1 in Volts
RANGRAW2=2 #PREGRED range for Channel 2 in Volts

# End of file AskN1997.INI

```

IV Aufstellung der verwendeten Batchdateien und Programme

Die folgende Aufstellung der verwendeten Batchdateien und Programme aus Kapitel 6 ist nach verschiedenen Gesichtspunkten geordnet. Die erste Einteilung erfolgt nach der Verwendung der Batchdateien (**blaue** und **grüne** Schriftfarbe) und Programme (**rote** Schriftfarbe) im Verlauf der Datenauswertung. Zudem sind über die Ordnung der Ebenen und der Verzeichnisnamen (schwarze Schriftfarbe) der Ort der Batchdateien und deren Aufrufe skizziert. Werden von den entsprechenden Batchdateien bzw. Programmen Eingabeparameter verlangt so sind diese mit angegeben. Für die Gedankenstriche sind die auf derselben Ebene befindlichen Programme einzusetzen.

```
\Eterna34\Auswertung-KTB
  ausgepackt_verschieben.bat
  DatAdd.bat
    DatAdd.exe
  Preanalyse.exe
  EternaAdd.bat
  Start.bat
    \1_Berg
      Berg.bat
        Filter.exe
        Kalibrierung.exe
        Transformation.exe
        ChFormat.exe
        Analyze.exe
        ResultCut.exe
        CutLowPass.exe
    \2_Eiglasdorf
      Eiglasdorf.bat
      -
    \3_Stockau
      Stockau.bat
      -
    \4_Mittelberg
      Mittelberg.bat
      -
    \5_Puellersreuth
      Puellersreuth.bat
      -
  StartTeil-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
    \1_Berg
      TeilAnalyseBerg-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
        PartCut.exe [Zeitmarke1] [Zeitmarke2]
        ChFormat.exe
        ModifyIni.exe [Zeitmarke1]
        Analyze.exe
        ResultCut.exe
    \2_Eiglasdorf
      TeilAnalyseEiglasdorf-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
      -
    \3_Stockau
      TeilAnalyseStockau-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
      -
```







```

\4_Mittelberg
    TeilAnalyseMittelberg-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
-
\5_Puellersreuth
    TeilAnalysePuellersreuth-Entwicklung.bat [Zeitmarke1] [Zeitmarke2]
-
StartTeil-Variation.bat [Zeitmarke1] [Zeitmarke2]
\1_Berg
    TeilAnalyseBerg-Variation.bat [Zeitmarke1] [Zeitmarke2]
    PartCut.exe [Zeitmarke1] [Zeitmarke2]
    ChFormat.exe
    ModifyIni.exe [Zeitmarke1]
    Analyze.exe
    ResultCut.exe
\2_Eiglasdorf
    TeilAnalyseEiglasdorf-Variation.bat [Zeitmarke1] [Zeitmarke2]
-
\3_Stockau
    TeilAnalyseStockau-Variation.bat [Zeitmarke1] [Zeitmarke2]
-
\4_Mittelberg
    TeilAnalyseMittelberg-Variation.bat [Zeitmarke1] [Zeitmarke2]
-
\5_Puellersreuth
    TeilAnalysePuellersreuth-Variation.bat [Zeitmarke1] [Zeitmarke2]
-
Start-KTB.bat
\1_Berg
    Berg-KTB.bat
    Transformation.exe
    ChFormat.exe
    Analyze.exe
    ResultCut.exe
    CutLowPass.exe
\2_Eiglasdorf
    Eiglasdorf-KTB.bat
-
\3_Stockau
    Stockau-KTB.bat
-
\4_Mittelberg
    Mittelberg-KTB.bat
-
\5_Puellersreuth
    Puellersreuth-KTB.bat
-
StartTeil-KTB.bat [Zeitmarke1] [Zeitmarke2]
\1_Berg
    TeilAnalyseBerg-KTB.bat [Zeitmarke1] [Zeitmarke2]
    PartCut.exe [Zeitmarke1] [Zeitmarke2]
    ChFormat.exe
    ModifyIni.exe [Zeitmarke1]
    Analyze.exe
    ResultCut.exe
\2_Eiglasdorf
    TeilAnalyseEiglasdorf-KTB.bat [Zeitmarke1] [Zeitmarke2]
-
\3_Stockau
    TeilAnalyseStockau-KTB.bat [Zeitmarke1] [Zeitmarke2]
-
\4_Mittelberg
    TeilAnalyseMittelberg-KTB.bat [Zeitmarke1] [Zeitmarke2]
-
5_Puellersreuth
    TeilAnalysePuellersreuth-KTB.bat [Zeitmarke1] [Zeitmarke2]
-

```

V Aufstellung der vorhandenen Neigungsmesser- und Grundwasserdaten

Legende:

X	X-Kanal der jeweiligen Station	Station:
Y	Y-Kanal der jeweiligen Station	B Berg
G	Grundwasserdaten	E Eiglasdorf
	Daten für entsprechenden Tag nicht vorhanden	S Stockau
	Daten für entsprechenden Tag des Neigungsmessers vorhanden	M Mittelberg
	Daten für entsprechenden Tag des Divers vorhanden	P Püllersreuth
	Daten für entsprechenden Tag nicht in Ordnung	

jul.	Datum	Wo.-	B			E			S			M			P		
Tag	tag	tag	X	Y	G	X	Y	G	X	Y	G	X	Y	G	X	Y	G
305	1.11.2003	Sa															
306	2.11.2003	So															
307	3.11.2003	Mo															
308	4.11.2003	Di															
309	5.11.2003	Mi															
310	6.11.2003	Do															
311	7.11.2003	Fr															
312	8.11.2003	Sa															
313	9.11.2003	So															
314	10.11.2003	Mo															
315	11.11.2003	Di															
316	12.11.2003	Mi															
317	13.11.2003	Do															
318	14.11.2003	Fr															
319	15.11.2003	Sa															
320	16.11.2003	So															
321	17.11.2003	Mo															
322	18.11.2003	Di															
323	19.11.2003	Mi															
324	20.11.2003	Do															
325	21.11.2003	Fr															
326	22.11.2003	Sa															
327	23.11.2003	So															
328	24.11.2003	Mo															
329	25.11.2003	Di															
330	26.11.2003	Mi															
331	27.11.2003	Do															
332	28.11.2003	Fr															
333	29.11.2003	Sa															
334	30.11.2003	So															
335	1.12.2003	Mo															
336	2.12.2003	Di															
337	3.12.2003	Mi															
338	4.12.2003	Do															
339	5.12.2003	Fr															
340	6.12.2003	Sa															
341	7.12.2003	So															
342	8.12.2003	Mo															
343	9.12.2003	Di															
344	10.12.2003	Mi															
345	11.12.2003	Do															
346	12.12.2003	Fr															
347	13.12.2003	Sa															
348	14.12.2003	So															
349	15.12.2003	Mo															
350	16.12.2003	Di															
351	17.12.2003	Mi															
352	18.12.2003	Do															
353	19.12.2003	Fr															

jul.	Datum	Wo.-	B			E			S			M			P		
Tag	tag	tag	X	Y	G	X	Y	G	X	Y	G	X	Y	G	X	Y	G
43	12.2.2004	Do															
44	13.2.2004	Fr															
45	14.2.2004	Sa															
46	15.2.2004	So															
47	16.2.2004	Mo															
48	17.2.2004	Di															
49	18.2.2004	Mi															
50	19.2.2004	Do															
51	20.2.2004	Fr															
52	21.2.2004	Sa															
53	22.2.2004	So															
54	23.2.2004	Mo															
55	24.2.2004	Di															
56	25.2.2004	Mi															
57	26.2.2004	Do															
58	27.2.2004	Fr															
59	28.2.2004	Sa															
60	29.2.2004	So															
61	1.3.2004	Mo															
62	2.3.2004	Di															
63	3.3.2004	Mi															
64	4.3.2004	Do															
65	5.3.2004	Fr															
66	6.3.2004	Sa															
67	7.3.2004	So															
68	8.3.2004	Mo															
69	9.3.2004	Di															
70	10.3.2004	Mi															
71	11.3.2004	Do															
72	12.3.2004	Fr															
73	13.3.2004	Sa															
74	14.3.2004	So															
75	15.3.2004	Mo															
76	16.3.2004	Di															
77	17.3.2004	Mi															
78	18.3.2004	Do															
79	19.3.2004	Fr															
80	20.3.2004	Sa															
81	21.3.2004	So															
82	22.3.2004	Mo															
83	23.3.2004	Di															
84	24.3.2004	Mi															
85	25.3.2004	Do															
86	26.3.2004	Fr															
87	27.3.2004	Sa															
88	28.3.2004	So															
89	29.3.2004	Mo															
90	30.3.2004	Di															
91	31.3.2004	Mi															

Jul.	Datum	Wo.-	B			E			S			M			P			Jul.	Datum	Wo.-	B			E			S			M			P		
Tag		tag	X	Y	G	X	Y	G	X	Y	G	X	Y	G	X	Y	G	Tag		tag	X	Y	G	X	Y	G	X	Y	G	X	Y	G			
196	14.7.2004	Mi																275	1.10.2004	Fr															
197	15.7.2004	Do																276	2.10.2004	Sa															
198	16.7.2004	Fr																277	3.10.2004	So															
199	17.7.2004	Sa																278	4.10.2004	Mo															
200	18.7.2004	So																279	5.10.2004	Di															
201	19.7.2004	Mo																280	6.10.2004	Mi															
202	20.7.2004	Di																281	7.10.2004	Do															
203	21.7.2004	Mi																282	8.10.2004	Fr															
204	22.7.2004	Do																283	9.10.2004	Sa															
205	23.7.2004	Fr																284	10.10.2004	So															
206	24.7.2004	Sa																285	11.10.2004	Mo															
207	25.7.2004	So																286	12.10.2004	Di															
208	26.7.2004	Mo																287	13.10.2004	Mi															
209	27.7.2004	Di																288	14.10.2004	Do															
210	28.7.2004	Mi																289	15.10.2004	Fr															
211	29.7.2004	Do																290	16.10.2004	Sa															
212	30.7.2004	Fr																291	17.10.2004	So															
213	31.7.2004	Sa																292	18.10.2004	Mo															
214	1.8.2004	So																293	19.10.2004	Di															
215	2.8.2004	Mo																294	20.10.2004	Mi															
216	3.8.2004	Di																295	21.10.2004	Do															
217	4.8.2004	Mi																296	22.10.2004	Fr															
218	5.8.2004	Do																297	23.10.2004	Sa															
219	6.8.2004	Fr																298	24.10.2004	So															
220	7.8.2004	Sa																299	25.10.2004	Mo															
221	8.8.2004	So																300	26.10.2004	Di															
222	9.8.2004	Mo																301	27.10.2004	Mi															
223	10.8.2004	Di																302	28.10.2004	Do															
224	11.8.2004	Mi																303	29.10.2004	Fr															
225	12.8.2004	Do																304	30.10.2004	Sa															
226	13.8.2004	Fr																305	31.10.2004	So															
227	14.8.2004	Sa																306	1.11.2004	Mo															
228	15.8.2004	So																307	2.11.2004	Di															
229	16.8.2004	Mo																308	3.11.2004	Mi															
230	17.8.2004	Di																309	4.11.2004	Do															
231	18.8.2004	Mi																310	5.11.2004	Fr															
232	19.8.2004	Do																311	6.11.2004	Sa															
233	20.8.2004	Fr																312	7.11.2004	So															
234	21.8.2004	Sa																313	8.11.2004	Mo															
235	22.8.2004	So																314	9.11.2004	Di															
236	23.8.2004	Mo																315	10.11.2004	Mi															
237	24.8.2004	Di																316	11.11.2004	Do															
238	25.8.2004	Mi																317	12.11.2004	Fr															
239	26.8.2004	Do																318	13.11.2004	Sa															
240	27.8.2004	Fr																319	14.11.2004	So															
241	28.8.2004	Sa																320	15.11.2004	Mo															
242	29.8.2004	So																321	16.11.2004	Di															
243	30.8.2004	Mo																322	17.11.2004	Mi															
244	31.8.2004	Di																323	18.11.2004	Do															
245	1.9.2004	Mi																324	19.11.2004	Fr															
246	2.9.2004	Do																325	20.11.2004	Sa															
247	3.9.2004	Fr																326	21.11.2004	So															
248	4.9.2004	Sa																327	22.11.2004	Mo															
249	5.9.2004	So																328	23.11.2004	Di															
250	6.9.2004	Mo																329	24.11.2004	Mi															
251	7.9.2004	Di																330	25.11.2004	Do															
252	8.9.2004	Mi																331	26.11.2004	Fr															
253	9.9.2004	Do																332	27.11.2004	Sa															
254	10.9.2004	Fr																333	28.11.2004	So															
255	11.9.2004	Sa																334	29.11.2004	Mo															
256	12.9.2004	So																335	30.11.2004	Di															
257	13.9.2004	Mo																336	1.12.2004	Mi															
258	14.9.2004	Di																337	2.12.2004	Do															
259	15.9.2004	Mi																338	3.12.2004	Fr															
260	16.9.2004	Do																339	4.12.2004	Sa															
261	17.9.2004	Fr																340	5.12.2004	So															
262	18.9.2004	Sa																341	6.12.2004	Mo															
263	19.9.2004	So																342	7.12.2004	Di															
264	20.9.2004	Mo																343	8.12.2004	Mi															
265	21.9.2004	Di																344	9.12.2004	Do															
266	22.9.2004	Mi																345	10.12.2004	Fr															
267	23.9.2004	Do																346	11.12.2004	Sa															
268	24.9.2004	Fr																347	12.12.2004	So															
269	25.9.2004	Sa											</																						

Anhang V: Aufstellung der vorhandenen Neigungsmesser- und Grundwasserdaten

Jul.	Datum	Wo.-tag	B			E			S			M			P		
Tag	tag		X	Y	G	X	Y	G	X	Y	G	X	Y	G	X	Y	G
354	19.12.2004	So															
355	20.12.2004	Mo															
356	21.12.2004	Di															
357	22.12.2004	Mi															
358	23.12.2004	Do															
359	24.12.2004	Fr															
360	25.12.2004	Sa															
361	26.12.2004	So															
362	27.12.2004	Mo															
363	28.12.2004	Di															
364	29.12.2004	Mi															
365	30.12.2004	Do															
366	31.12.2004	Fr															
1	1.1.2005	Sa															
2	2.1.2005	So															
3	3.1.2005	Mo															
4	4.1.2005	Di															
5	5.1.2005	Mi															
6	6.1.2005	Do															
7	7.1.2005	Fr															
8	8.1.2005	Sa															
9	9.1.2005	So															
10	10.1.2005	Mo															
11	11.1.2005	Di															
12	12.1.2005	Mi															
13	13.1.2005	Do															
14	14.1.2005	Fr															
15	15.1.2005	Sa															
16	16.1.2005	So															
17	17.1.2005	Mo															
18	18.1.2005	Di															
19	19.1.2005	Mi															
20	20.1.2005	Do															
21	21.1.2005	Fr															
22	22.1.2005	Sa															
23	23.1.2005	So															
24	24.1.2005	Mo															
25	25.1.2005	Di															
26	26.1.2005	Mi															
27	27.1.2005	Do															
28	28.1.2005	Fr															
29	29.1.2005	Sa															
30	30.1.2005	So															
31	31.1.2005	Mo															
32	1.2.2005	Di															
33	2.2.2005	Mi															
34	3.2.2005	Do															
35	4.2.2005	Fr															
36	5.2.2005	Sa															
37	6.2.2005	So															
38	7.2.2005	Mo															
39	8.2.2005	Di															
40	9.2.2005	Mi															
41	10.2.2005	Do															
42	11.2.2005	Fr															
43	12.2.2005	Sa															
44	13.2.2005	So															
45	14.2.2005	Mo															
46	15.2.2005	Di															
47	16.2.2005	Mi															
48	17.2.2005	Do															
49	18.2.2005	Fr															
50	19.2.2005	Sa															
51	20.2.2005	So															
52	21.2.2005	Mo															
53	22.2.2005	Di															
54	23.2.2005	Mi															
55	24.2.2005	Do															
56	25.2.2005	Fr															
57	26.2.2005	Sa															
58	27.2.2005	So															
59	28.2.2005	Mo															
60	1.3.2005	Di															
61	2.3.2005	Mi															
62	3.3.2005	Do															
63	4.3.2005	Fr															
64	5.3.2005	Sa															
65	6.3.2005	So															
66	7.3.2005	Mo															
67	8.3.2005	Di															
68	9.3.2005	Mi															
69	10.3.2005	Do															
70	11.3.2005	Fr															
71	12.3.2005	Sa															
72	13.3.2005	So															
73	14.3.2005	Mo															
74	15.3.2005	Di															
75	16.3.2005	Mi															
76	17.3.2005	Do															
77	18.3.2005	Fr															
78	19.3.2005	Sa															
79	20.3.2005	So															
80	21.3.2005	Mo															
81	22.3.2005	Di															
82	23.3.2005	Mi															
83	24.3.2005	Do															
84	25.3.2005	Fr															
85	26.3.2005	Sa															
86	27.3.2005	So															
87	28.3.2005	Mo															
88	29.3.2005	Di															
89	30.3.2005	Mi															
90	31.3.2005	Do															
91	1.4.2005	Fr															
92	2.4.2005	Sa															
93	3.4.2005	So															
94	4.4.2005	Mo															
95	5.4.2005	Di															
96	6.4.2005	Mi															
97	7.4.2005	Do															
98	8.4.2005	Fr															
99	9.4.2005	Sa															
100	10.4.2005	So															
101	11.4.2005	Mo															
102	12.4.2005	Di															
103	13.4.2005	Mi															
104	14.4.2005	Do															
105	15.4.2005	Fr															
106	16.4.2005	Sa															
107	17.4.2005	So															
108	18.4.2005	Mo															
109	19.4.2005	Di															
110	20.4.2005	Mi															
111	21.4.2005	Do															
112	22.4.2005	Fr															
113	23.4.2005	Sa															
114	24.4.2005	So															
115	25.4.2005	Mo															
116	26.4.2005	Di															
117	27.4.2005	Mi															
118	28.4.2005	Do															

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Jena, 01.03.2006

André Gebauer

